

Assistability

Software Design Project

By: SSA-21 Dream Team

Table of Contents

Executive Summary.....	10
Project Plan	11
User Requirements	12
User Stories.....	12
Client User Stories.....	12
Caregiver User Stories.....	13
Administrator User Stories.....	14
Use Case Write-Ups	15
Client Use Cases.....	15
Select Adult User.....	15
Create Account.....	17
Login to Existing Account.....	19
Edit Existing Account.....	21
View Attainment Goal.....	23
Child opens company website	24
Option to Sign In	26
User has chosen to create a new child account.....	28
Choosing adult or child account.....	30
Creating an account	32
Linking the child account to an adult account, or adult email.....	34
Remember Password functionality.....	36
Children's User Interface	38
Children's Picture Tutorial	40
Start Learning Goal	42
Going through a Goal.....	44
Reward System	46
Delete Account.....	48
Create a Journal	50
Reading Journal Posts	52
Delete Journal	54
Open Program.....	56

Choosing adult or child account.....	58
Learning Goals.....	60
Anxiety Example Goal	62
User Achievement Earn	64
Deleting Goal.....	66
Create a Journal	68
Reading Journal Posts	69
Delete Journal	70
Create Extinction Goal (Administrator).....	71
Create Extinction Goal (Client).....	73
Edit Extinction Goal (Administrator).....	75
Edit Extinction Goal (Client)	77
Assign Extinction Goal.....	79
Assign Extinction Goal (Client)	81
Update Extinction Goal (Administrator)	83
Update Extinction Goal (Client)	85
View Extinction Goal (Administrator)	87
View Extinction Goal (Caregiver)	88
View Extinction Goal (Client)	89
View Completed Goals (Administrator)	90
View Completed Goals (Caregiver)	91
View Completed Goals (Client)	92
Remove Extinction Goal (Administrator)	93
Remove Extinction Goal (Client)	94
Deactivate Extinction Goal (Administrator)	95
Deactivate Extinction Goal (Client)	97
View Goal Details	99
Create Incident (Administrator).....	100
Create Incident (Caregiver)	102
Create Incident (Client)	104
Edit Incident (Administrator)	106
Edit Incident (Client)	108
Edit Incident (Caregiver)	110

Assign Incident (Administrator)	112
Assign Incident (Client)	114
Assign Incident (Caregiver)	116
View Incident (Administrator).....	118
View Incident (Client).....	119
View Incident (Caregiver).....	120
Track Incidents Over Time (Administrator).....	121
Track Incidents Over Time (Client).....	122
Track Incidents Over Time (Caregiver).....	123
See Report of Incidents Based on Date (Administrator).....	124
See Report of Incidents Based on Date (Client).....	125
See Report of Incidents Based on Date (Caregiver).....	126
See Dates Based on Type/Name of Incident (Administrator).....	127
See Dates Based on Type/Name of Incident (Client).....	128
See Dates Based on Type/Name of Incident (Caregiver).....	129
See Details of Specific Incident (Administrator)	130
See Details of Specific Incident (Client)	131
See Details of Specific Incident (Caregiver)	133
Record Incident (Administrator)	134
Record Incident (Client)	136
Record Incident (Caregiver)	138
Create Journal	140
Create Journal Entry.....	141
Create Profile	142
Create Profile	143
Delete Journal Entry.....	144
Delete Profile	145
Delete Profile	146
Retrieve Journal Entries	147
Retrieve Profile	148
Retrieve Profile	149
Update Journal Entry	150
Update Profile	151

Update Profile	152
Create Journal	153
Create Journal Entry.....	154
Create Profile	155
Create Profile	156
Delete Journal Entry.....	157
Delete Learning Goals	158
Delete Profile	159
Retrieve Journal Entries	160
Retrieve Profile	161
Retrieve Profile	162
Caregiver Use Cases	163
Deactivate Own Account	163
Delete Account.....	165
Update Account	167
Deny Given Permissions.....	169
Creating Account.....	171
Forgot Login Information.....	173
Leave Note on Wall	175
User Login	177
User Logout.....	179
Reactivate Note.....	181
Request Permissions	183
Edit Note	185
Leave a Note on a Task	187
Deactivate Note	189
Delete Note.....	191
Deny Invitation.....	193
Accept Invitation.....	195
Check Off Items From Daily Task List.....	197
View All Notifications.....	199
View Tasks by Topic	200
View Client's Current Tasks.....	202

Deny Given Permissions.....	204
Reactivate Account	206
Refresh Screen	208
Remove Client.....	209
View All Activities.....	211
Administrative Use Cases.....	213
Create Account.....	213
Add Client.....	215
Remove Client.....	217
Add Caregiver.....	218
Remove Caregiver	220
Edit Caregiver Permissions.....	222
Create Permissions Profile	224
Deactivate Account.....	226
Log In.....	227
Use Case Drawings.....	229
Client Processing Space.....	229
Caregiver Processing Space.....	230
Administrative Processing Space	231
User Interface Drawings	232
Login and Account Creation Interfaces.....	232
Caregiver Interfaces	250
Client Interfaces	282
Administrative Interfaces.....	312
Data Layer	344
Implementation Assumptions.....	344
E-R Diagram.....	345
Data Dictionary	346
Data Operations – Stored Procedure Definitions	349
User	349
Password	350
Login.....	350
Role	350

Group	351
Membership.....	351
Membership Role.....	352
Journal.....	352
Journal Entry	353
Message	353
Invitation.....	354
Invitation Type	355
Acceptance.....	355
Routine.....	356
Routine Completion	356
Routine Step.....	357
Routine Step Completion.....	358
Habitual Goal	358
Extinction Goal.....	359
Incident	360
Incident Event	361
Attainment Goal.....	363
Performance	363
Performance Event	364
Award.....	366
Award Achievement.....	367
Reward	367
Reward Status	368
Domain Models.....	369
System Diagrams: Data Flows	373
Messaging System.....	373
Messaging	373
Invitation System	374
Invitations	374
Journal System	375
Making a Journal.....	375
Making a Journal Entry.....	375

Editing a Journal.....	377
Editing a Journal Entry	378
Viewing Journals and Entries	379
Goal System	380
Create Goal	380
Routine System	380
Complete Step/Routine	381
Complete Step Routine Context Diagram.....	382
UML Class Diagrams.....	383
Data Access Layer	383
User.....	383
Message	383
Membership.....	383
Role	384
Invitation.....	384
Journal.....	384
JournalEntry	385
Incident	385
Performance	385
AttainmentGoal.....	386
ExtinctionGoal.....	386
HabitualGoal	387
GoalType	387
Routine.....	387
RoutineStep.....	388
RoutineStepCompletion.....	388
RoutineCompletion.....	388
Award.....	389
Reward	389
Logic Layer.....	390
User.....	390
Message	390
Membership.....	390

Role	391
Invitation.....	391
Journal.....	391
JournalEntry	391
Incident	392
Performance	392
AttainmentGoal.....	392
ExtinctionGoal.....	393
HabitualGoal	393
GoalType	393
Routine.....	394
RoutineStep.....	394
RoutineStepCompletion.....	394
RoutineCompletion.....	395
Award.....	395
Reward.....	395
Additional Processing Models.....	396
Activity Diagrams	396
User to User Messaging	396
Goal progress viewing, setting, and receiving	397
Sequence Diagrams.....	398
State Chart Diagrams	399
Invitations	399
Awards	400
Goals	401
Assign/Change User Roles.....	402
Deployment Plan.....	403
Project Methodology	403
Development Technologies	404
Target Deployment Technologies	405
Programming Team.....	405

Executive Summary

By Ryan Taylor, other contributors: Becky Baenziger, William Clark, Whitney Vinson

This application is designed to aid individuals in building routines, setting goals, and monitoring behaviors. This will allow a person or their guardian to create a goal-based reward system. Our goal is help users with a wide range of behavioral, cognitive, and physical disabilities by incentivizing desired outcomes. Our user demographic will span all ages, from young children to seniors.

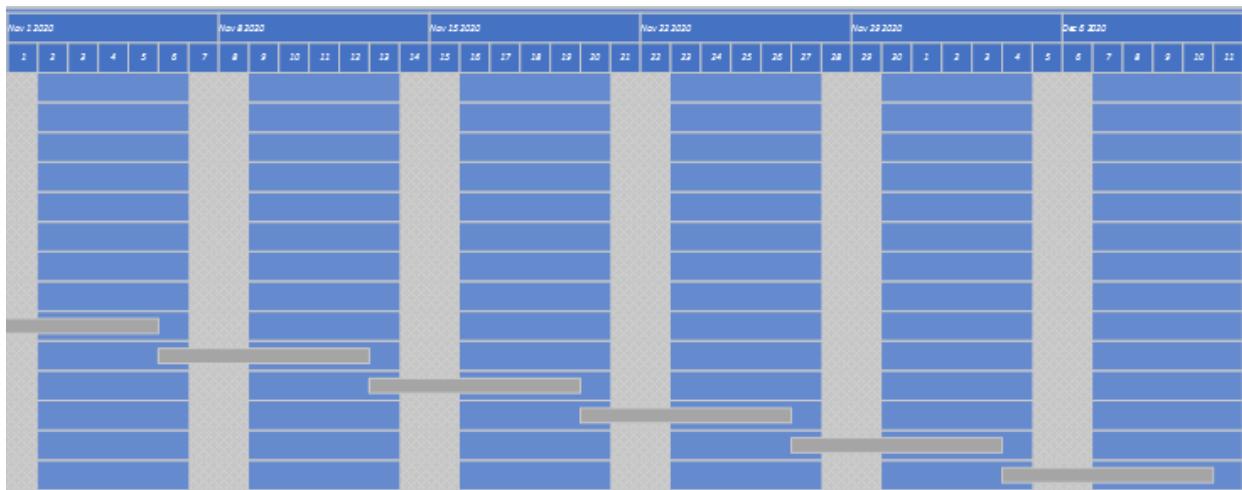
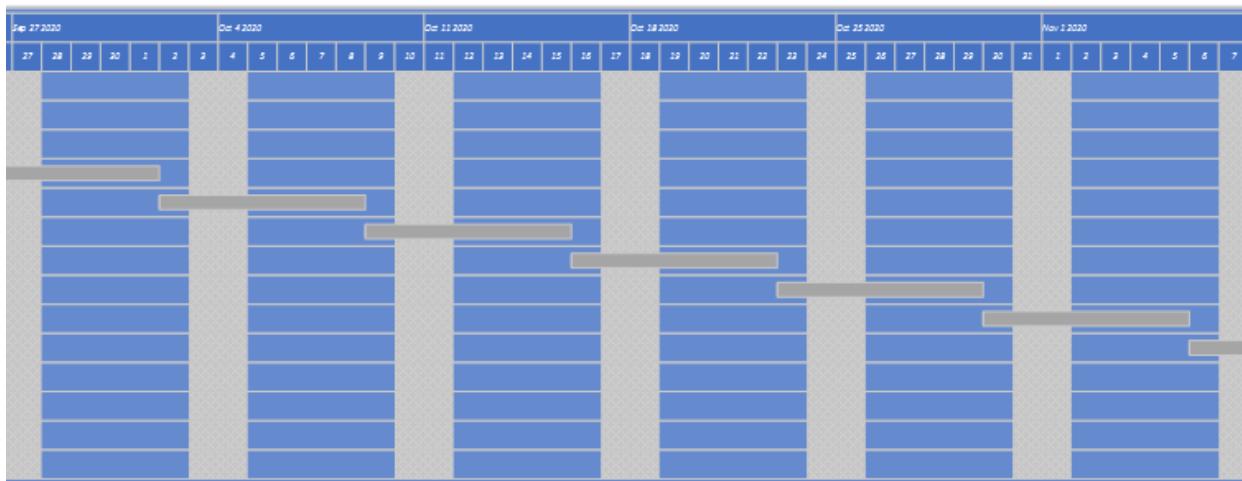
Individual users will be able to create and monitor their own goals and routines, mark their completion, track achievements and redeem rewards. Caregivers within a user's group will be able to view user goals to aid in their achievement. They will have the ability to join a group, monitor progress, add comments, drop a client, request permissions, and mark progress on goals. Messages can be exchanged between group members, dependent on permission.

The administrative role will have the ability to create and implement goals for users. They will have the ability to catalog achievements and the status of rewards, and create groups including other caregivers. Administrators will also be able to grant and revoke permissions from group members as needed.

Project Plan

By Whitney Vinson

ID	Task Name	Start	Finish	Duration	Aug 2020				Sep 1 2020				Sep 12 2020				Sep 20 2020									
					4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	Kickoff Week – Assigning Roles	9/4/2020	9/10/2020	1w																						
2	Sprint 1 – User Stories	9/11/2020	9/17/2020	1w																						
3	Sprint 1 – Use Cases	9/18/2020	9/24/2020	1w																						
4	Sprint 1 – User Interface	9/25/2020	10/1/2020	1w																						
5	Review Week	10/2/2020	10/8/2020	1w																						
6	Sprint 2 – E-R Diagrams	10/9/2020	10/15/2020	1w																						
7	Sprint 2 – UML Models (Storage/View)	10/16/2020	10/22/2020	1w																						
8	Sprint 2 – UML Models (Data Access Classes/Stored Procedures)	10/23/2020	10/29/2020	1w																						
9	Review Week	10/30/2020	11/5/2020	1w																						
10	Sprint 3 – Activity/Sequence Diagrams	11/6/2020	11/12/2020	1w																						
11	Sprint 3 – Data Flow/Statechart Diagrams	11/13/2020	11/19/2020	1w																						
12	Sprint 3 – Class/Object Inheritance Diagrams	11/20/2020	11/26/2020	1w																						
13	Review Week	11/27/2020	12/3/2020	1w																						
14	Finalization	12/4/2020	12/10/2020	1w																						



User Requirements

By Nick Loesel, contributions by: Becky Baenziger, William Clark, Mitchell Paul, Ryan Taylor, Whitney Vinson, Nathaniel Webber, Jory Wernette

User Stories

Client User Stories

As a client I need to create an account in order to make a profile.

As a client I need to login in order to access the program's features.

As a client I need to edit my profile in order to update my profile.

As a client I need to delete my account in order to remove my information.

As a client I need to view a conversation in order to interact with other users.

As a client I need to send a message in order to message other users.

As a client I need to send a message to a group in order to message a group of users.

As a client I need to create a journal in order to record my thoughts/give feedback.

As a client I need to update a journal in order to add additional thoughts/ give more feedback.

As a client I need to view a journal in order to see the thoughts / feedback.

As a client I need to delete a journal in order to remove thoughts/ feedback.

As a client I need to create a learning goal in order to create a personalized goal.

As a client I need to view a learning goal in order to track progress on that goal.

As a client I need to delete a learning goal in order to delete a completed goal.

As a client I need to update a learning goal in order to better fit my needs.

As a client I need to create an extinction goal in order to create a personalized goal.

As a client I need to view an extinction goal in order to track progress on that goal.

As a client I need to delete an extinction goal in order to delete a completed goal.

As a client I need to update an extinction goal in order to better fit my needs.

As a client I need to complete a goal in order to work towards completing my routine.

As a client I need to create a routine in order to see what goes I need to complete for the day.

As a client I need to view my routine in order to see my progress.

As a client I need to delete a routine in order to delete a completed routine.

As a client I need to update a routine in order to better fit my needs.

Caregiver User Stories

As a caregiver I need to create an account in order to make a profile.

As a caregiver I need to login in order to access the program's features.

As a caregiver I need to edit my profile in order to update my profile.

As a caregiver I need to delete my account in order to remove my information.

As a caregiver I need to view a conversation in order to message other users.

As a caregiver I need to send a message in order to message other users.

As a caregiver I need to send a message to a group in order to message a group of users.

As a caregiver I need to create a journal in order to record my thoughts/give feedback.

As a caregiver I need to update a journal in order to add additional thoughts/ give more feedback.

As a caregiver I need to view a journal in order to see the thoughts / feedback.

As a caregiver I need to delete a journal in order to remove thoughts/ feedback.

As a caregiver I need to create a learning goal in order to create a personalized goal.

As a caregiver I need to view a learning goal in order to track progress on that goal.

As a client I need to delete a learning goal in order to delete a completed goal.

As a caregiver I need to update a learning goal in order to better fit their needs.

As a caregiver I need to create an extinction goal in order to create a personalized goal.

As a caregiver I need to view an extinction goal in order to track progress on that goal.

As a client I need to delete an extinction goal in order to delete a completed goal.

As a caregiver I need to update an extinction goal in order to better fit their needs.

As a caregiver I need to accept an invite in order to become a caregiver.

As a caregiver I need to view client list in order to better manage my clients.

As a caregiver I need to create a routine in order to see what my client needs to complete for the day.

As a caregiver I need to view my routine in order to see my progress.

As a caregiver I need to update a routine in order to better fit my client's needs.

As a caregiver I need to delete a routine in order to delete a completed routine.

Administrator User Stories

By William Clark, Nick Loesel

As an administrator I need to create an account in order to make a profile.

As an administrator I need to login in order to access the program's features.

As an administrator I need to edit my profile in order to update my profile.

As an administrator I need to delete my account in order to remove my information.

As an administrator I need to view a conversation in order to message other users.

As an administrator I need to send a message in order to message other users.

As an administrator I need to send a message to a group in order to message a group of users.

As an administrator I need to create a journal in order to record my thoughts/give feedback.

As an administrator I need to update a journal in order to add additional thoughts/ give more feedback.

As an administrator I need to view a journal in order to see the thoughts / feedback.

As an administrator I need to delete a journal in order to remove thoughts/ feedback.

As an administrator I need to create a learning goal in order to create a personalized goal.

As an administrator I need to view a learning goal in order to track progress on that goal.

As a client I need to delete a learning goal in order to delete a completed goal.

As an administrator I need to update a learning goal in order to better fit their needs.

As an administrator I need to create an extinction goal in order to create a personalized goal.

As an administrator I need to view an extinction goal in order to track progress on that goal.

As a client I need to delete an extinction goal in order to delete a completed goal.

As an administrator I need to update an extinction goal in order to better fit their needs.

As an administrator I need to invite other users in order to make them a caregiver.

As an administrator I need to view client list in order to better manage my clients.

Use Case Write-Ups

By Mitchell Paul, contributions by: Becky Baenziger, William Clark, Nick Loesel, Ryan Taylor, Whitney Vinson, Nathaniel Webber, Jory Wernette

Client Use Cases

Select Adult User

Use Case ID:	1		
Use Case Name:	Select Adult User		
Created By:	Whitney Black	Last Updated By:	Nick Loesel
Date Created:	2020-09-22	Last Revision Date:	2020-12-11
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	Select option to set-up a new adult user account.		
Trigger:	A user clicks on select adult user		
Preconditions:	1. Administrator must be logged in to account 2. Must be an administrator		
Postconditions:	1. Client becomes adult user		
Normal Flow:	1. Administrator selects the select adult user button 2. Administrator selects which client they would like to be an adult user 3. The Interface asks the user if they're sure they want this person to be an adult user 4. Administrator clicks yes 5. The program displays a success message		
Alternative Flows: [Alternative Flow 1 – User Selects Incorrect Option]	3. Administrator selects the wrong user a. Administrator clicks no to selecting this user b. Program returns to basic flow step 2		
Exceptions:	N/A		
Includes:	Account Set-Up		
Frequency of Use:	On Demand		

Special Requirements:	None
Assumptions:	The client has an account The Administrator has administrative privileges
Notes and Issues:	N/A

Create Account

Use Case ID:	2		
Use Case Name:	Create Account		
Created By:	Whitney Black	Last Updated By:	Nick Loesel
Date Created:	2020-09-22	Last Revision Date:	12/11/20
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	User creates an account		
Trigger:	User selects setup for Adult User Account		
Preconditions:	1. User has a computer		
Postconditions:	1. User creates an account		
Normal Flow:	1. User enters first name 2. User enters last name 3. User enters a new username. 4. System validates that username fits username criteria 5. System validates that username does not already exist 6. User enters a new password 7. System validates that password fits security standards 8. User re-enters password 9. System validates that the password lines match 10. User enters birthdate 11. User selects disability type 12. User selects disability 13. User selects create new account		
Alternative Flows: [Alternative Flow 1 – Blank Fields]	3. Username doesn't fit criteria <ol style="list-style-type: none"> The system will prompt the user to change username The system will focus cursor in the username field Program returns to basic flow step 2 5. Username already exists <ol style="list-style-type: none"> The system will prompt the user to change username The system will focus cursor in the username field Program returns to basic flow step 2 6. Password isn't strong enough <ol style="list-style-type: none"> The system will prompt the user to change password The system will focus cursor in the password field Program returns to basic flow step 6 9. Passwords don't match <ol style="list-style-type: none"> The system will prompt the user to change password The system will focus cursor in the matching password field Program returns to basic flow step 2 		
Exceptions:	N/A		
Includes:	Account set-up		

Frequency of Use:	On Demand
Special Requirements:	Username and password criteria set
Assumptions:	User knows how to set up an account
Notes and Issues:	Dropdown box to select between account setups in create account feature and making initial main screen a login in/sign up screen.

Login to Existing Account

Use Case ID:	3		
Use Case Name:	Login to Existing Account		
Created By:	Whitney Black	Last Updated By:	Nick Loesel
Date Created:	2020-09-22	Last Revision Date:	2020-12-11
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	User can sign into a user account that retains their data.		
Trigger:	Client wants to use their profile in the program.		
Preconditions:	1. Client must have a user account previously set up. 2. Client remembers or has access to password.		
Postconditions:	1. Client can successfully log into the user profile. 2. Client can access their information.		
Normal Flow:	1. User selects existing user option. 2. System prompts client to enter username. 3. System prompts client to enter password. 4. User enters username. 5. User enters password. 6. User submits information with login button. 7. System checks for matching username. 8. System checks for matching password. 9. User is allowed into their existing account.		
Alternative Flows: [Alternative Flow 1 – Blank Fields]	4. User has a wrong username <ol style="list-style-type: none"> The system will alert the user that the username is incorrect. The system will prompt user to select forgot username. The user will be sent the username to their email address on file. System returns to basic flow step 4 5. User enters the wrong password three times <ol style="list-style-type: none"> The system will alert the user that the password is incorrect. The system will prompt the user to select reset password. The user will be sent a link to reset their password System returns to basic flow step 2 9. User leaves a field blank <ol style="list-style-type: none"> The system will prompt the client to fill in all fields The system will focus cursor in first empty field. System returns to basic flow step 4 		

Exceptions:	4. User enters an unknown username a. Username is invalid. b. Message to customer to re-enter username. c. Customer enters correct username. d. Use Case resumes on step 5 of normal flow. 6. enters an incorrect password a. Password is invalid. b. Message to client to re-enter password. c. Client enters correct password. d. Use Case resumes on step 6.
Includes:	User login
Frequency of Use:	On demand
Special Requirements:	Password Hash/Validator
Assumptions:	The user already has an account.
Notes and Issues:	None

Edit Existing Account

Use Case ID:	4		
Use Case Name:	Edit Existing Account		
Created By:	Whitney Black	Last Updated By:	Whitney Black
Date Created:	2020-09-24	Last Revision Date:	12/11/2020
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	The ability to edit account information.		
Trigger:	Client wants to edit account		
Preconditions:	1. Client must have a user account previously set up. 2. Client remembers or has access to password.		
Postconditions:	1. Client can successfully log into the user profile. 2. Client can access their information.		
Normal Flow:	1. Client selects existing user option. 2. System prompts client to enter username. 3. System prompts client to enter password. 4. Client enters username. 5. Client enters password. 6. Client submits information with login button. 7. System checks for matching username. 8. System checks for matching password. 9. Client is allowed into their existing account. 10. Client selects Edit Profile option. 11. Client edits profile information. 12. Client saves new profile information.		
Alternative Flows:	11a. In step 11 of the normal flow, if the client leaves a field blank		
[Alternative Flow 1 – Blank Entry]	1. System will replace blank with original information.		
Exceptions:	N/A		
Includes:	Steps 1-12 in the normal flow.		
Frequency of Use:	As needed.		
Special Requirements:	N/A		
Assumptions:	Client can successfully log into the user profile. Client can access their information.		

Notes and Issues:	N/A
--------------------------	-----

View Attainment Goal

Use Case ID:	5		
Use Case Name:	View Attainment Goal		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/21/2020	Last Revision Date:	2020-12-11
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	User views a Learning Goal		
Trigger:	User wants to view a Learning Goal		
Normal Flow:	1. User selects "View Learning Goal" 2. System Prompts User for Name of Goal 3. User types Name of Goal 4. System prompts User for Type of Goal 5. User enters Type of Goal 6. System Displays Goals		
Frequency of Use:	On Demand		
Special Requirements:	N/A		
Assumptions:	That the child can type or has access software that can help with typing. That the student has a web browser. That the student has access to internet That the student has a web browsing device		
Notes and Issues:	1. Will the homepage look the same if the user has an account, compared to a non-user?		

Child opens company website

Use Case ID:	6		
Use Case Name:	Child opens company website		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/21/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	To access the website the child must use their tablet, enter the domain in their web browser, press enter		
Trigger:	Child is told to go to the website or decides on his own		
Preconditions:	1. Child must have a web browsing device. 2. Child must have a web browser		
Postconditions:	1. Child arrives on websites homepage. 2. Child arrives on homepage while logged in if password is remembered.		
Normal Flow:	1. Child boots up web browsing devices 2. Child selects web browser 3. Child enters or uses a search engine to find the website 4. Child Selects website		
Alternative Flows: [Alternative Flow 1 – Not in Network]			

Exceptions:	2a. In step 2 of the normal flow, if the customer enters and the website is down 1. Display an error message about what is going on 2. Communicate through other services that our systems are down
Includes:	
Frequency of Use:	On Demand
Special Requirements:	Internet connection
Assumptions:	That the child can type or has access software that can help with typing. That the student has a web browser. That the student has access to internet That the student has a web browsing device
Notes and Issues:	2. Will the homepage look the same if the user has an account, compared to a non-user?

Option to Sign In

Use Case ID:	7		
Use Case Name:	Option to sign-in		
Created By:	Mitchell Paul	Last Updated By:	
Date Created:	9/21/2020	Last Revision Date:	
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	When the home page is visited the user is given an option to sign-in.		
Trigger:	Opening of the webpage.		
Preconditions:	Webpage must be displayed		
Postconditions:	1. Child will either be signed in or redirected to the account creation page.		
Normal Flow:	1. Login screen is displayed 2. User enters information 3. If correct, signed in 4. If not, error displays 5. If no sign in information, given option to create account.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	4a. In step 4 of the normal flow, if the customer is not in the website database 1. System will prompt user to create account 2. Customer accepts		

	3. Use Case resumes on step 3
Exceptions:	<p>2a. In step 2 of the normal flow, if the customer enters an invalid Account</p> <p>3. Sign in is not allowed</p> <p>4. Message to customer to re-enter Username and password</p> <p>5. Customer enters correct Username and password</p> <p>6. Use Case resumes on step 3 of normal flow</p>
Includes:	1a Opening the webpage.
Frequency of Use:	On demand.
Special Requirements:	
Assumptions:	<p>That the child can type or has access software that can help with typing.</p> <p>That the student has a web browser.</p> <p>That the student has access to internet</p> <p>That the student has a web browsing device</p>
Notes and Issues:	N/A

User has chosen to create a new child account

Use Case ID:	8		
Use Case Name:	User has chosen to create a new child account		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	09/24/2020	Last Revision Date:	2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Users need to be able to create accounts.		
Trigger:	1. User is at the website homepage. 2. Does not have an account 3. Chooses the option to create an account.		
Preconditions:	1. User does not have an account 2. User wants an account 3. User Clicks on create an account.		
Postconditions:	2. User is brought to the create a creation page.		
Normal Flow:	5. User is on the homepage 6. User wants an account, does not have account 7. Clicks “Create an account” 8. Brought to the create an account page.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	5a User does not want an account 5b User leaves website		

Includes:	1a Opening the website 1b Option to sign-in
Frequency of Use:	Whenever an account is being created
Special Requirements:	
Assumptions:	That the child can type or has access software that can help with typing. That the student has a web browser. That the student has access to internet That the student has a web browsing device
Notes and Issues:	<ul style="list-style-type: none"> • Do we want a version of the website that can be used without an account, to the users with privacy concerns?

Choosing adult or child account

Use Case ID:	9		
Use Case Name:	Choosing adult or child account		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:			Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Choosing between adult and child account		
Trigger:	After the child enters the create account screen they are presented with a choice of an adult or child account.		
Preconditions:	3. The child wants to create an account.		
Postconditions:	3. The account is set as a child or adult account, whatever is selected.		
Normal Flow:	1. Child enters account creation screen. 2. Given the choice of adult or child 3. Benefits and drawbacks of both presented 4. Child selects child account 5. Moves on into a tailored environment for a child to create an account.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	2. Child accidentally selects adult account <ol style="list-style-type: none"> If the child regrets this decision, easy UI to return If the child or guardian wants an adult account a confirmation box pops up and then they pass through. 		

Exceptions:	1. Select the wrong account type 2. Easy UI to return should be implemented.
Includes:	5, 6 , 7 , 8
Frequency of Use:	Whenever an account is created
Special Requirements:	Easy to read and understand, easy to backtrack.
Assumptions:	Adult supervision while creating an online account, if they cannot be verified the child account will take such little information to create it should not be a big concern
Notes and Issues:	Extremely easy to use UI at this point would be much more useful than a more complicated, extensive UI.

Creating an account

Use Case ID:	10		
Use Case Name:	Creating an account		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:			Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	To create an account, to track progress and access the website features.		
Trigger:	<ul style="list-style-type: none"> • User selects ***** "Create a child account." ***** 		
Preconditions:	<ul style="list-style-type: none"> • User selected, Create a child account 		
Postconditions:	4. After this process is completed the child will have a full functional child's account 5. Allowing them access to the websites features while also having a more child friendly experience.		
Normal Flow:	1. User is redirected to an account creation screen. 2. Asks for a username and does active referencing of the database to avoid username creation errors 3. Asks for a parent or guardians' email for connecting, and password resets 4. Asks for a password 5. Asks for confirmation of password 6. Asks for age 7. Asks for a first name or nickname 8. These are all that are needed for a basic children's account.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	2a User enters a username already in use, prompts for different name until unique name is given 4a Different passwords are entered, clears both fields and allows the password to appear in plaintext for easier matching.		

Exceptions:	<ul style="list-style-type: none"> • Username already in use, prompts for a different one • Incorrect password, clears entries, plaintext.
Includes:	6, 7, 8
Frequency of Use:	Whenever an account is created.
Special Requirements:	Less information needed the better.
Assumptions:	<ul style="list-style-type: none"> • Parent has an email. Student or parent can type. • Child can set up a username and password.
Notes and Issues:	<ul style="list-style-type: none"> • Not sure if the child should be able to create an account without a parent's email, discussion about that.

Linking the child account to an adult account, or adult email.

Use Case ID:	11		
Use Case Name:	Linking the child account to an adult account, or adult email.		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Administrator Secondary Actor 2. DBMS 3. Interface		
Description:	Linking the child account to the adult account		
Trigger:	Child account is created, email not given on sign-up. Parent now wants a link to the child's account.		
Preconditions:	<ul style="list-style-type: none"> • Child account created • Parent has email 		
Postconditions:	<ul style="list-style-type: none"> • Child account is now linked to parents 		
Normal Flow:	1. Sign into child's account 2. Click settings 3. Link accounts 4. Type in email 5. Confirm on parents' email.		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:	<ul style="list-style-type: none"> • Incorrect email entered, displays all currently linked emails. • Easy to remove access. • Change privileges for different emails. 		
Includes:	10		
Frequency of Use:	<ul style="list-style-type: none"> • Whenever the child or parents wants someone else to have access to the account • The access is determined by the privileges given. 		

Assumptions:	<ul style="list-style-type: none">• Child account created• Parent has email
Notes and Issues:	<ul style="list-style-type: none">• Additional functionality for a connection between an adult account and a child account.

Remember Password functionality

Use Case ID:	12		
Use Case Name:	Remember Password functionality		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/22/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	<ul style="list-style-type: none"> After an account is created, allow a “remember password option” on home screen 		
Trigger:	<ul style="list-style-type: none"> Account is created Checkbox selected labeled remember password 		
Preconditions:	<ul style="list-style-type: none"> Account created 		
Postconditions:	<ul style="list-style-type: none"> Password is saved in web browser Easy login next visit 		
Normal Flow:	<ol style="list-style-type: none"> User opens webpage with account already created Types in Username and password correctly Checkbox below that states “Remember Password?” Click checkbox Next time you visit website username and password already filled in Ready to login 		
Alternative Flows: [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> User incorrectly types username or password, gives them an error they try again. 		
Exceptions:	<ul style="list-style-type: none"> Incorrect password entered Entry in password box deleted Password must be resubmitted, hopefully correctly After three bad attempts prompt for password recovery system 		
Includes:	The rest of step 1.		
Frequency of Use:	After clicking the checkbox, it would be saved until unchecked		

Special Requirements:	
Assumptions:	<ul style="list-style-type: none"> • User has password • User understands clicking a checkbox enables it
Notes and Issues:	<ul style="list-style-type: none"> • Gives user an explanation of what the checkbox does the first time it is clicked. As children might not have been exposed to a feature like this before.

Children's User Interface

Use Case ID:	13		
Use Case Name:	Children's User Interface		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/23/20		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Once the child has logged into the website, they are presented with a child friendly interface		
Trigger:	<ul style="list-style-type: none"> • Selecting children's account in the account creation options • Logging in 		
Preconditions:	<ul style="list-style-type: none"> • Children's account 		
Postconditions:	<ul style="list-style-type: none"> • Child friendly UI increases understanding and participation • Encourages the child • Makes them want to return to the website and improve further 		
Normal Flow:	<ol style="list-style-type: none"> 1. Child creates account 2. Child Logs in 3. Presented with a simple UI 		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:	<ol style="list-style-type: none"> 1. Child fails to log in 2. Never presented with children friendly UI 		
Includes:	Account creation		
Frequency of Use:	Every time the child logs in to their account		

Special Requirements:	<ul style="list-style-type: none"> • Child friendly UI • Encouraging colors • Simple words and explanations
Assumptions:	<ul style="list-style-type: none"> • Child has made an account • Child would prefer the child UI on their first login • Can be changed if the child would prefer a fresh look or system
Notes and Issues:	<ul style="list-style-type: none"> • Research into what a child friendly UI would look like • What colors work best • What fonts work best • Which explanations work best.

Children's Picture Tutorial

Use Case ID:	14		
Use Case Name:	Children's Picture Tutorial		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/20		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	After the child's first log in they are presented with a picture tutorial on how to work the website		
Trigger:	<ul style="list-style-type: none"> • Childs first login 		
Preconditions:	<ul style="list-style-type: none"> • Children's account • Has not been logged in before 		
Postconditions:	<ul style="list-style-type: none"> • Hopefully, child has a better understanding of how to work the website • Can be returned to at any point 		
Normal Flow:	<ol style="list-style-type: none"> 1. Child logs in for the first time 2. Start of picture tutorial starts 3. Works through all the features 4. Can be repeated at any point 5. Once completed the first Goal should be started and ready to be used 		
Alternative Flows: [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> 1. Child accidentally skips through or does not understand 2. Simple tutorial button will be left on the screen somewhere for easy relearning 		
Exceptions:	<p>[Describe any anticipated error conditions that could occur during execution of the use case and define how the system is to respond to those conditions.</p> <p>e.g., Exceptions to the Withdraw Case transaction</p> <p>2a. In step 2 of the normal flow, if the customer enters an invalid PIN</p> <p>7. Transaction is disapproved</p> <p>8. Message to customer to re-enter PIN</p> <p>9. Customer enters correct PIN</p> <p>10. Use Case resumes on step 3 of normal flow]</p>		

Includes:	<ul style="list-style-type: none"> • Create an account • Login
Frequency of Use:	<ul style="list-style-type: none"> • First Login • Whenever the user chooses to try again
Special Requirements:	<ul style="list-style-type: none"> • Simple • Audio version • Picture Version • Text version • Or all three
Assumptions:	<ul style="list-style-type: none"> • User has account • User can see • User can hear • User can read • One of these would be a requirement
Notes and Issues:	<ol style="list-style-type: none"> 1. Tutorial could be interactive 2. Or just highlighting while an audio plays along with subtitles to explain the program.

Start Learning Goal

Use Case ID:	15		
Use Case Name:	Start Learning Goal		
Created By:	Mitchell	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Presents the child with three choices to work on		
Trigger:	<ul style="list-style-type: none"> After the tutorial is completed this is where you are guided to 		
Preconditions:	<ul style="list-style-type: none"> Children's account Tutorial completed 		
Postconditions:	<ul style="list-style-type: none"> Child is on a self-guided lesson plan 		
Normal Flow:	<ol style="list-style-type: none"> Tutorial is completed Child is left with three large buttons on skills to work on with descriptions Physical – Physical ailments Behavioral – Behavioral ailments Mental – Mental Once one of those is chosen, they are brought to a screen with many different choices of skills to work on Recommend that adults help with these choices 		
Alternative Flows: [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> Child leaves website after the tutorial Take them to NF 2 		
Exceptions:	<ul style="list-style-type: none"> Child mis clicks on the wrong path Have easy to access "back" button 		
Includes:	<ul style="list-style-type: none"> Account creation Tutorial 		

Frequency of Use:	Whenever a user would like to switch between paths
Special Requirements:	<ul style="list-style-type: none"> • Large • Easy to read • Colorful • Descriptions
Assumptions:	<ul style="list-style-type: none"> • Child can get adult help while setting up • Child or adult can decide what type of help they need •
Notes and Issues:	<ul style="list-style-type: none"> • Maybe change the name of paths to something less bold • Easy to understand for a small child description should be written

Going through a Goal

Use Case ID:	16		
Use Case Name:	Going through a Goal		
Created By:	Mitchell	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	The average Goal experience		
Trigger:	<ul style="list-style-type: none"> • Child and adult in conjunction select a Goal 		
Preconditions:	<ul style="list-style-type: none"> • Goal is chosen 		
Postconditions:	<ul style="list-style-type: none"> • Child learns a skill or lesson that helps them with their day-to-day life • Child reviews a subject to reaffirm a skill or lesson in their day-to-day life 		
Normal Flow:	<ol style="list-style-type: none"> 1. After a path is chosen 2. Child beings 3. Simple pictures 4. Simple videos 5. Simple audio clips 6. Are all played to convey a well-researched, age-appropriate message. 7. Easy, encouraging quizzes and assignments are spread throughout 		
Alternative Flows: [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> 1. Child does not understand <ol style="list-style-type: none"> a. Child can retake lesson or take a similar one b. Child cannot pass the lesson c. Child is suggested a much easier, confidence building assignment 		
Exceptions:	<ol style="list-style-type: none"> 1. Internet could go out, lesson is brought back to the last known point 2. Child becomes overwhelmed, easy button to end lesson there and to come back to it later. 		
Includes:	<ul style="list-style-type: none"> • Account Creation • Goals 		
Frequency of Use:	Every time a Goal is used		

Special Requirements:	Child friendly, easy to understand, multitude of lessons
Assumptions:	<ul style="list-style-type: none"> • Child can read • Could implement some lessons to help with young readers • Child can sit through a short lesson
Notes and Issues:	<ul style="list-style-type: none"> • Must be very encouraging and easy to start • But not too easy as to bore the child • Some minigames to help convey messages

Reward System

Use Case ID:	17		
Use Case Name:	Reward System		
Created By:	Mitchell	Last Updated By:	
Date Created:	9/24/2020	Last Revision Date:	
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Rewarding the child for completing paths, with a point system		
Trigger:	<ul style="list-style-type: none"> • Completing Paths • Competing Assignments • Correct or mostly correct quiz answers 		
Preconditions:	<ul style="list-style-type: none"> • Children's account • Working through paths 		
Postconditions:	<ul style="list-style-type: none"> • Child is given a report of points earned • Can be redeemed at parents' discretion • Gives a sense of achievement 		
Normal Flow:	<ol style="list-style-type: none"> 1. Goal, assignment, or Quiz is completed 2. Points are distributed 3. Child is rewarded 		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:	<ul style="list-style-type: none"> • Child is not earning enough • Child is earning too much • Adult can adjust the point rates 		
Includes:	<ul style="list-style-type: none"> • Goals 		
Frequency of Use:	<ul style="list-style-type: none"> • After every quiz, assignment, Goal completed 		
Special Requirements:	<ul style="list-style-type: none"> • Theme can specify the reward currency i.e., coins, flowers, shells 		

Assumptions:	<ul style="list-style-type: none">• Parent would reward their children's progression through the program to encourage more use of it
Notes and Issues:	N/A

Delete Account

Use Case ID:	18		
Use Case Name:	Delete Account		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	If the user or adult is not happy with the program their account can be deleted.		
Trigger:	<ul style="list-style-type: none"> • Settings – Delete Account 		
Preconditions:	<ul style="list-style-type: none"> • Account must be created 		
Postconditions:	<ul style="list-style-type: none"> • Account is deleted 		
Normal Flow:	<ol style="list-style-type: none"> 1. Settings tab holds the delete account menu 2. If the account is linked to an email, upon deletion it will send an email that will be used to confirm the deletion. 3. If the account is not linked. The user will have 48 hours (about 2 days) to relogging to stop the deletion of the account from frustration 		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:	<ul style="list-style-type: none"> • Account does not delete • Try again • No E-mail Received • Try again or contact support 		
Includes:	<ul style="list-style-type: none"> • Account Creation • Account Deletion 		
Frequency of Use:	<ul style="list-style-type: none"> • Upon deletion of account 		

Special Requirements:	<ul style="list-style-type: none">• Access to email
Assumptions:	<ul style="list-style-type: none">• Many frustration deletes will happen• Some methods to reduce those should be instilled
Notes and Issues:	

Create a Journal

Use Case ID:	19		
Use Case Name:	Create a Journal		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Providing the ability to create a journal		
Trigger:	<ul style="list-style-type: none"> • “Create a journal” 		
Preconditions:	<ul style="list-style-type: none"> • User account is created • Linked with parent account 		
Postconditions:	<ul style="list-style-type: none"> • Journal is created with a link for parents or caregivers to read 		
Normal Flow:	<ol style="list-style-type: none"> 1. Click on “Create a Journal” 2. Choose a name 3. Choose a color 4. Begin your first entry 		
Alternative Flows: [Alternative Flow 1 – Not in Network]	N/A		
Exceptions:	Database Exception		
Includes:	<ul style="list-style-type: none"> • Account creation • Parent Email 		
Frequency of Use:	Whenever a child would like to create a journal post		
Special Requirements:	<ul style="list-style-type: none"> • Different Themes • Emailed reports of journals if parent does not have account 		

Assumptions:	<ul style="list-style-type: none">• Student would like to journal• Student would like to share journal with parent
Notes and Issues:	N/A

Reading Journal Posts

Use Case ID:	20		
Use Case Name:	Reading Journal Posts		
Created By:	Mitchell Paul	Last Updated By:	
Date Created:	9/24/2020		Last Revision Date:
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Sharing and reading journal posts		
Trigger:	<ul style="list-style-type: none"> • After a journal has been created and shared • The journal can be read by those given permission to read it 		
Preconditions:	<ul style="list-style-type: none"> • Journal Has been created with a post 		
Postconditions:	<ul style="list-style-type: none"> • Those given permission to read the posts, can read them at their discretion while they are still holding permission to view the posts 		
Normal Flow:	<ol style="list-style-type: none"> 1. After a post is created 2. The ability to share it is revealed 3. Emails can be typed in or accounts linked can view anytime 4. Or they can be set as private and only let you view them on the account that created them 		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:			
Includes:	<ul style="list-style-type: none"> • Journal Creation 		

Frequency of Use:	Whenever journals are created
Special Requirements:	<ul style="list-style-type: none"> • Privacy • Permissions • View who read it maybe
Assumptions:	<ul style="list-style-type: none"> • Children would be okay with some reading their journals • Who you would want to read would change entry to entry?
Notes and Issues:	

Delete Journal

Use Case ID:	21		
Use Case Name:	Delete Journal		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Deleting a journal post		
Trigger:	Clicking the Delete Journal Post button		
Preconditions:	<ul style="list-style-type: none"> • Account has been created • Journal has been created • Journal post has been created 		
Postconditions:	<ul style="list-style-type: none"> • Previously created journal post is now removed and unviewable 		
Normal Flow:	<ol style="list-style-type: none"> 1. Click on the journal post to deleted 2. Click the delete journal post button 		
Alternative Flows: [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> 1. Parent may read the post and want it deleted 2. Can access the child's account using connect email and delete the post themselves 		
Exceptions:	<ol style="list-style-type: none"> 1. Accidentally deleted post 2. Can be recovered through support 		
Includes:	<ul style="list-style-type: none"> • Journal Creation • Journal Post creation 		
Frequency of Use:	Whenever a post is deleted		
Special Requirements:	<ul style="list-style-type: none"> • Hold the post for 24 hours • Unviewable but can be recovered 		

Assumptions:	<ul style="list-style-type: none">Parents would read some posts, consider them inappropriate and would want to delete immediately.
Notes and Issues:	How long to store journals after “deletion “for?

Open Program

Use Case ID:	22		
Use Case Name:	Open Program		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Opening the program after installation		
Trigger:	<ul style="list-style-type: none"> • Double clicking the desktop icon • Clicking the executable • Clicking on the programs name in the start menu 		
Preconditions:	<ul style="list-style-type: none"> • Program has been installed from website 		
Postconditions:	<ul style="list-style-type: none"> • Program is now open on desktop displaying login and password fields 		
Normal Flow:	<ol style="list-style-type: none"> 1. Double click desktop icon 2. Short loading period 3. Program opens with login screen 		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:	<ol style="list-style-type: none"> 1. Program does not start 2. Try reinstalling 3. Contact support 4. Try the web browser version 		
Includes:			
Special Requirements:	<ul style="list-style-type: none"> • Creating an icon • Installer 		
Assumptions:	<ul style="list-style-type: none"> • User can figure out installer • User can figure out how to start program 		

Notes and Issues:	<ul style="list-style-type: none">• Installing should be as few clicks as possible• Program should not be large, should install quickly.• Install a desktop icon with installation
--------------------------	--

Choosing adult or child account

Use Case ID:	23		
Use Case Name:	Choosing adult or child account		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Choosing between adult and child account		
Trigger:	After the adult enters the create account screen they are presented with a choice of an adult or child account.		
Preconditions:	<ul style="list-style-type: none"> • The adult wants to create an account. 		
Postconditions:	<ul style="list-style-type: none"> • The account is set as a child or adult account, whatever is selected. 		
Normal Flow:	<ol style="list-style-type: none"> 1. Child enters account creation screen. 2. Given the choice of adult or child 3. Benefits and drawbacks of both presented 4. Adult selects adult account 5. Moves on into a tailored environment for an adult to create an account. 		
Alternative Flows: [Alternative Flow 1 – Not in Network]	<ol style="list-style-type: none"> 2. Adult accidentally selects child account <ol style="list-style-type: none"> a. If the adult regrets this decision, easy UI to return b. If the adult or guardian wants a child account a confirmation box pops up and then they pass through. 		
Exceptions:	<ol style="list-style-type: none"> 3. Select the wrong account type 4. Easy UI to return should be implemented. 		

Includes:	21
Frequency of Use:	Whenever an account is created
Special Requirements:	Easy to read and understand, easy to backtrack.
Assumptions:	Adult supervision while creating an online account, if they cannot be verified the child account will take such little information to create it should not be a big concern
Notes and Issues:	Extremely easy to use UI at this point would be much more useful than a more complicated, extensive UI.

Learning Goals

Use Case ID:	24		
Use Case Name:	Learning Goals		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	The average Goal experience		
Trigger:	<ul style="list-style-type: none"> • Adult and therapist in conjunction select a Goal 		
Preconditions:	<ul style="list-style-type: none"> • Goal is chosen 		
Postconditions:	<ul style="list-style-type: none"> • Child learns a skill or lesson that helps them with their day-to-day life • Child reviews a subject to reaffirm a skill or lesson in their day-to-day life 		
Normal Flow:	8. After a path is chosen 9. Adult 10. Depending on the lesson the format is different 11. Various levels and versions of most lessons 12. More focused on returning to normalcy as opposed to learning new skills 13. But new skills are available		
Alternative Flows: [Alternative Flow 1 – Not in Network]	1a Adult does not understand 1b Adult can retake lesson or take a similar one 1c Adult cannot pass the lesson 1d Adult is suggested a much easier, confidence building assignment		
Exceptions:	3. Internet could go out, lesson is brought back to the last known point 4. Child becomes overwhelmed, easy button to end lesson there and to come back to it later.		

Includes:	<ul style="list-style-type: none"> • Account Creation • Goals •
Frequency of Use:	Every time a Goal is used
Special Requirements:	Adult friendly, easy to understand, multitude of lessons
Assumptions:	<ul style="list-style-type: none"> • Adult can read • Could implement some lessons to help with young readers • Child can sit through a short lesson
Notes and Issues:	<ul style="list-style-type: none"> • Must be very encouraging and easy to start • But not too easy as to bore the adult • Some minigames to help convey messages

Anxiety Example Goal

Use Case ID:	25		
Use Case Name:	Anxiety Example Goal		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Gives an example of Goal that would deal with an anxiety case.		
Trigger:	<ul style="list-style-type: none"> • User selects anxiety Goal 		
Preconditions:	1. User selects mental 2. User selects Anxiety 3. User is ready to go through the Goal		
Postconditions:	1. User learns something about themselves 2. Coping methods 3. Why and what is happening		
Normal Flow:	1. User selects anxiety Goal 2. User clicks Begin Goal 3. User selects text-based learning 4. User is given a quick synopsis on triggers of anxiety and further resources to research 5. User is taught some coping methods 6. User is informed on physical symptoms 7. User is informed on when to seek professional help.		
Exceptions:	1. User leaves program early 2. Can be continued from their last known point 3. Or skipped through until they get to that point if they choose to.		
Includes:	<ul style="list-style-type: none"> • Create an account • Choose an ailment • Choose a Goal • Adult account 		
Frequency of Use:	<ul style="list-style-type: none"> • Whenever the anxiety Goal is chosen, could be often could be never 		

Special Requirements:	<ul style="list-style-type: none"> • Looks of informative information • Professionally researched information • Minimal homeopathic options
Assumptions:	<ul style="list-style-type: none"> • User can navigate simple UI • User is willing to read • User wants to be helped
Notes and Issues:	<ul style="list-style-type: none"> • With this being an adult account more descriptive language can be used • Easier to recommend help to an adult that is seeking help instead of a child that is being forced to complete the program.

User Achievement Earn

Use Case ID:	26		
Use Case Name:	User Achievement Earn		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020	Last Revision Date:	2020-12-11
Actors:	Primary Actor 1. User Secondary Actor 2. DBMS 3. Interface		
Description:	After a Goal is completed, the user is presented with an achievement badge		
Trigger:	<ul style="list-style-type: none"> • Completion of a Goal 		
Preconditions:	1. User has not previously completed the Goal		
Postconditions:	User Gains the badge for conquering that Goal.		
Normal Flow:	1. Presented to the user after successfully completing a Goal 2. Can be displayed on their profile and shared 3. Gives the user a sense of accomplishment and something to be proud of for their time spent improving themselves.		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:	1. Achievement or badge does not appear 2. Redo the Goal or contact support for the badge to be given to you		
Includes:			
Frequency of Use:	<ul style="list-style-type: none"> • After the completion of a Goal 		
Special Requirements:	<ul style="list-style-type: none"> • Unique badges and achievements must be created • Graphic design work 		
Assumptions:	<ul style="list-style-type: none"> • Users would be proud of their time spent improving themselves and would like a small thing to brag about. 		

Notes and Issues:	N/A
--------------------------	-----

Deleting Goal

Use Case ID:	27		
Use Case Name:	Deleting Goal		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Deleting the progress of a current goal to stop working on it or restart		
Trigger:	Clicking delete on the UI		
Preconditions:	1. Goal has been started and is saved at a point.		
Postconditions:	1. Goal is reset and can now be started again from the beginning.		
Normal Flow:	1. Goal was started one week 2. User is ten places in 3. Feels they do not remember much and wants a fresh slate 4. Goes to the menu page 5. Clicks the delete progress button on the far right 6. Confirmation window appears 7. Yes or No 8. User clicks yes 9. Goal Progress is deleted		
Alternative Flows: [Alternative Flow 1 – Not in Network]	7a user Accidentally clicks No, go through steps 1-6 again and selected Yes		
Exceptions:	<ul style="list-style-type: none"> • User tries to delete progress • Button Fails • Should be able to click back to the beginning of the lesson • Essentially performing the same thing as restarting 		
Includes:	<ul style="list-style-type: none"> • Create an Account • Select an ailment • Select a Goal • Delete a Goal 		
Frequency of Use:	When the user would like to restart or relearn certain paths		
Special Requirements:	UI delete button, confirmation window		

Assumptions:	<ul style="list-style-type: none">• User understands the delete would be permanent
Notes and Issues:	3. A 2.0 feature as the workaround of just clicking back would work well.

Create a Journal

Use Case ID:	28.		
Use Case Name:	Create a Journal		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Supplying the ability to create a journal		
Trigger:	<ul style="list-style-type: none"> • “Create a journal” 		
Preconditions:	<ul style="list-style-type: none"> • User account is created 		
Postconditions:	<ul style="list-style-type: none"> • Journal is created 		
Normal Flow:	5. Click on “Create a Journal” 6. Choose a name 7. Choose a color 8. Begin your first entry		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:			
Includes:	<ul style="list-style-type: none"> • Account creation 		
Frequency of Use:	Whenever an adult would like to create a journal post		
Special Requirements:	<ul style="list-style-type: none"> • Different Themes 		
Assumptions:	<ul style="list-style-type: none"> • Adult would like to journal • Adult would like to share journal with others 		
Notes and Issues:	What is the maximum size of the PIN that a user can have?		

Reading Journal Posts

Use Case ID:	29		
Use Case Name:	Reading Journal Posts		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Sharing and reading journal posts		
Trigger:	<ul style="list-style-type: none"> After a journal has been created and shared The journal can be read by those given permission to read it 		
Preconditions:	<ul style="list-style-type: none"> Journal Has been created with a post 		
Postconditions:	<ul style="list-style-type: none"> Those given permission to read the posts, can read them at their discretion while they are still holding permission to view the posts 		
Normal Flow:	<ul style="list-style-type: none"> After a post is created The ability to share it is revealed Emails can be typed in or accounts linked can view anytime Or they can be set as private and only let you view them on the account that created them 		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:			
Includes:	<ul style="list-style-type: none"> Journal Creation 		
Frequency of Use:	Whenever journals are created		
Special Requirements:	<ul style="list-style-type: none"> Privacy Permissions View who read it maybe 		
Assumptions:	<ul style="list-style-type: none"> The adult would be okay with someone reading their journals 		
Notes and Issues:			

Delete Journal

Use Case ID:	30		
Use Case Name:	Delete Journal		
Created By:	Mitchell Paul	Last Updated By:	Nick Loesel
Date Created:	9/24/2020		Last Revision Date: 2020-12-11
Actors:	Primary Actor 1. User 2. Caretaker Secondary Actor 2. DBMS 3. Interface		
Description:	Deleting a journal post		
Trigger:	Clicking the Delete Journal Post button		
Preconditions:	<ul style="list-style-type: none"> • Account has been created • Journal has been created • Journal post has been created 		
Postconditions:	<ul style="list-style-type: none"> • Previously created journal post is now deleted and unviewable 		
Normal Flow:	<ol style="list-style-type: none"> 1. Click on the journal post to deleted 2. Click the delete journal post button 		
Exceptions:	<ol style="list-style-type: none"> 1. Accidentally deleted post 2. Can be recovered through support 		
Includes:	<ul style="list-style-type: none"> • Journal Creation • Journal Post creation 		
Frequency of Use:	Whenever a post is deleted		
Special Requirements:	<ul style="list-style-type: none"> • Hold the post for 24 hours • Unviewable but can be recovered 		
Notes and Issues:	How long to store journals after "deletion "for?		

Create Extinction Goal (Administrator)

Use Case ID:	57	
Use Case Name:	Create Extinction Goal (Administrator)	
Created By:	Becky	Last Updated By:
Date Created:	2020-10-30	Last Revision Date:
Actors:	Primary Actor Administrator Secondary Actor(s) DBMS Interface	
Description:	To create Extinction goals for a client	
Trigger:	Administrator clicks on Create Goal	
Preconditions:	1. Administrator must be logged into account 2. Administrator must be on extinction goal interface	
Postconditions:	1. An extinction goal will have been created 2. An extinction goal will have been saved to the DBMS	
Normal Flow:	1. Administrator clicks on create goal button 2. Administrator enters data in fields/form 3. Administrator saves 4. Interface sends saved goal to DBMS 5. DBMS stores goal appropriately	
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Administrator corrects data. Return to normal flow step 2. 2. Failed to Save Goal (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Return to normal flow step 3 3. Failed to save goal (exit) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Exit goal creation interface. 	
Exceptions:		

Includes:	
Frequency of Use:	On demand
Special Requirements:	
Assumptions:	
Notes and Issues:	<ol style="list-style-type: none">1) What are data constraints on fields necessary for creating the goal2) Ways to have options to choose from when creating goal

Create Extinction Goal (Client)

Use Case ID:	58		
Use Case Name:	Create Extinction Goal (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary: 1. DBMS 2. Interface		
Description:	To create extinction goals for the client		
Trigger:	Client clicks on create goal		
Preconditions:	1. Client must be logged into account 2. Client must be on extinction goal interface		
Postconditions:	1. An extinction goal will have been created 2. An extinction goal will have been saved to the DBMS		
Normal Flow:	1. Client clicks on create goal button 2. Client enters data in fields/form 3. Client saves 4. Interface sends saved goal to DBMS 5. DBMS stores goal appropriately		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Client corrects data. Return to normal flow step 2. 2. Failed to Save Goal (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to client Interface asks if want to try to save again or exit Return to normal flow step 3 3. Failed to save goal (exit) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to client Interface asks if want to try to save again or exit Exit goal creation interface. 		
Exceptions:			
Includes:			

Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	<ul style="list-style-type: none"> • What are data constraints on fields necessary for creating the goal • Ways to have options to choose from when creating goal

Edit Extinction Goal (Administrator)

Use Case ID:	59		
Use Case Name:	Edit Extinction Goal (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor - Administrator Secondary Actor(s) – DBMS, Interface		
Description:	Administrator can Edit the Extinction Goal		
Trigger:	Administrator clicks on Edit button		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at Extinction Goal interface		
Postconditions:	1. An extinction goal will have been edited 2. An extinction goal will have been saved to the DBMS		
Normal Flow:	1. Administrator selects goal to be edited 2. Administrator clicks on edit goal button 3. Administrator enters data in fields/form 4. Administrator saves 5. Interface sends saved goal to DBMS 6. DBMS stores goal appropriately		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Administrator corrects data. Return to normal flow step 2. 2. Failed to Save Goal (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Return to normal flow step 3 3. Failed to save goal (exit) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Exit edit goal interface. 		
Exceptions:			
Includes:			
Frequency of Use:	On demand		

Special Requirements:	
Assumptions:	

Edit Extinction Goal (Client)

Use Case ID:	60		
Use Case Name:	Edit Extinction Goal (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 2. Client Secondary: 3. DBMS 4. Interface		
Description:	To edit extinction goals by the client		
Trigger:	Client clicks on edit goal		
Preconditions:	1. Client must be logged into account 2. Client must be on extinction goal interface		
Postconditions:	1. An extinction goal will have been edited 2. An extinction goal will have been saved to the DBMS		
Normal Flow:	1. Client selects goal to be edited 2. Client clicks on Edit Goal button 3. Client enters data in fields/form 4. Client saves 5. Interface sends saved goal to DBMS 6. DBMS stores goal appropriately		
Alternative Flows:	1. Invalid Data Entry a) Invalid data entered (normal flow step 2) b) Interface sends error message to screen c) Client corrects data. d) Return to normal flow step 2. 2. Failed to Save Goal (select SAVE again) a) DBMS cannot save goal (normal flow step 4) b) Interface sends error message to client c) Interface asks if want to try to save again or exit d) Return to normal flow step 3 2. Failed to save goal (exit) a. DBMS cannot save goal (normal flow step 4) b. Interface sends error message to client c. Interface asks if want to try to save again or exit d. Exit edit goal interface.		
Exceptions:			

Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	<ul style="list-style-type: none"> • What are data constraints on fields necessary for editing the goal • Ways to have options to choose from when editing goal

Assign Extinction Goal

Use Case ID:	61		
Use Case Name:	Assign Extinction Goal		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	<p>Primary Actor</p> <ol style="list-style-type: none">1. Administrator <p>Secondary:</p> <ol style="list-style-type: none">1. DBMS2. Interface		
Description:	To assign extinction goals to the client		
Trigger:	Administrator clicks on an Assign Goal		
Preconditions:	<ol style="list-style-type: none">1. Administrator must be logged into account2. Administrator must be at Extinction Goal interface		
Postconditions:	<ol style="list-style-type: none">1. A goal will be assigned to a client2. A goal assignment will have been saved to the DBMS		
Normal Flow:	<ol style="list-style-type: none">1. Administrator selects/clicks on client2. Administrator selects goal to be added3. Administrator drags to/selects client4. Administrator clicks save5. Interface sends saved assigned goal to DBMS6. DBMS stores goal appropriately		
Alternative Flows:	<ol style="list-style-type: none">1. Invalid Data Entry<ol style="list-style-type: none">a. Invalid data entered (normal flow step 2)b. Interface sends error message to screenc. Administrator corrects data.d. Return to normal flow step 2.2. Failed to Assign Goal (select SAVE again)<ol style="list-style-type: none">a. DBMS cannot assign goal (normal flow step 4)b. Interface sends error message to Administratorc. Interface asks if want to try to assign goal again or exitd. Return to normal flow step 3		

	<p>3. Failed to assign goal (exit)</p> <ul style="list-style-type: none"> a. DBMS cannot assign goal (normal flow step 4) b. Interface sends error message to Administrator c. Interface asks if want to try to assign goal again or exit d. Exit assign goal interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	<ul style="list-style-type: none"> 1. Will it be a dragging to/selecting/clicking client situation? 2. Ways to have options to assign multiple goals at a time

Assign Extinction Goal (Client)

Use Case ID:	62		
Use Case Name:	Assign Extinction Goal (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	<p>Primary Actor</p> <ol style="list-style-type: none">1. Client <p>Secondary:</p> <ol style="list-style-type: none">2. DBMS3. Interface		
Description:	To assign extinction goals to the client		
Trigger:	Client clicks on goal		
Preconditions:	<ol style="list-style-type: none">1. Client must be logged into account2. Client must be at goal interface		
Postconditions:	<ol style="list-style-type: none">1. A goal will be assigned to a client2. A goal assignment will have been saved to the DBMS		
Normal Flow:	<ol style="list-style-type: none">1. Client clicks on assign goal2. Client drags to/selects goal3. Client clicks save4. Interface sends saved assigned goal to DBMS5. DBMS stores goal appropriately		

Alternative Flows:	<ol style="list-style-type: none"> 1. Invalid Data Entry <ol style="list-style-type: none"> a. Invalid data entered (normal flow step 2) b. Interface sends error message to screen c. Client corrects data. d. Return to normal flow step 2. 2. Failed to Assign Goal (select SAVE again) <ol style="list-style-type: none"> a. DBMS cannot assign goal (normal flow step 4) b. Interface sends error message to Client c. Interface asks if want to try to assign goal again or exit d. Return to normal flow step 3 3. Failed to Assign goal (exit) <ol style="list-style-type: none"> a. DBMS cannot assign goal (normal flow step 4) b. Interface sends error message to Client c. Interface asks if want to try to assign goal again or exit d. Exit assign goal interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	<ol style="list-style-type: none"> 1. Will it be a dragging to/selecting/clicking client situation? 2. Ways to have options to assign multiple goals at a time

Update Extinction Goal (Administrator)

Use Case ID:	63		
Use Case Name:	Update Extinction Goal (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	<p>Primary Actor Administrator</p> <p>Secondary: DBMS Interface</p>		
Description:	To update extinction goal for a client		
Trigger:	Administrator clicks on an update goal		
Preconditions:	<ul style="list-style-type: none">Administrator must be logged into accountAdministrator must be at extinction goal interface		
Postconditions:	<ul style="list-style-type: none">A goal will be updated by an AdministratorA goal update will have been saved to the DBMS		
Normal Flow:	<ol style="list-style-type: none">Administrator selects clientAdministrator selects goal to be updatedAdministrator modifies goalAdministrator clicks saveInterface sends saved updated goal to DBMSDBMS stores goal appropriately		
Alternative Flows:	<ol style="list-style-type: none">Invalid Data Entry<ol style="list-style-type: none">Invalid data entered (normal flow step 2)Interface sends error message to screenAdministrator corrects data.Return to normal flow step 2.Failed to Save Update Goal (select SAVE again)<ol style="list-style-type: none">DBMS cannot save goal (normal flow step 4)Interface sends error message to AdministratorInterface asks if want to try to save again or exitReturn to normal flow step 3		

	<p>2. Failed to save update goal (exit)</p> <ul style="list-style-type: none"> a. DBMS cannot save goal (normal flow step 4) b. Interface sends error message to Administrator c. Interface asks if want to try to save again or exit d. Exit goal creation interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	Goal update can just be marking completed

Update Extinction Goal (Client)

Use Case ID:	64		
Use Case Name:	Update Extinction Goal (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30		Last Revision Date:
Actors:	Primary Actor 1. Client Secondary Actor(s) 1. DBMS 2. Interface		
Description:	To update extinction goal by the client		
Trigger:	Client clicks on update goal		
Preconditions:	1. Client must be logged into account 2. Client must be at extinction goal interface		
Postconditions:	1. A goal will be updated by a client 2. A goal update will have been saved to the DBMS		
Normal Flow:	1. Client selects on goal to be updated 2. Client modifies goal 3. Client clicks save goal 4. Interface sends saved assigned goal to DBMS 5. DBMS stores goal appropriately		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Client corrects data. Return to normal flow step 2. 2. Failed to Save updated Goal (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Client Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save updated goal (exit)</p> <p>3. DBMS cannot save goal (normal flow step 4)</p> <ul style="list-style-type: none"> b. Interface sends error message to client c. Interface asks if want to try to save again or exit d. Exit update goal interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	<p>1. Goal update can just be marking goal completed</p>

View Extinction Goal (Administrator)

Use Case ID:	65		
Use Case Name:	View Extinction Goal (Administrator)		
Created By:	Becky	Last Updated By:	Whitney Vinson
Date Created:	2020-10-30	Last Revision Date:	12/11/2020
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 1. DBMS 2. Interface		
Description:	View details of extinction goal		
Trigger:	Administrator double clicks on goal Administrator highlights goal and clicks view		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at goal interface		
Postconditions:	1. A goals details will be viewed by the administrator		
Normal Flow:	1. Administrator double-clicks on goal to be viewed 2. Interface displays goal to administrator 3. Administrator clicks exit		
Alternative Flows:	1. Failed to display goal a. Interface could not display goal (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to goal interface		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Will want an edit, exit, assign options?		

View Extinction Goal (Caregiver)

Use Case ID:	66		
Use Case Name:	View Extinction Goal (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary: 1. DBMS 2. Interface		
Description:	View details of extinction goal		
Trigger:	Caregiver double-clicks on goal Caregiver highlights goal and clicks view		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at goal interface		
Postconditions:	1. A goals details will be viewed by the caregiver		
Normal Flow:	1. Caregiver double-clicks on goal to be viewed 2. Interface displays goal to administrator 3. Caregiver clicks exit		
Alternative Flows:	2. Failed to display goal a. Interface could not display goal (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to goal interface		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Will want an edit, exit, assign options?		

View Extinction Goal (Client)

Use Case ID:	67		
Use Case Name:	View Extinction Goal (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 1. DBMS 2. Interface		
Description:	View details of extinction goal		
Trigger:	1. Client double-clicks on goal 2. Client highlights goal and clicks view		
Preconditions:	1. Client must be logged into account 2. Client must be at goal interface		
Postconditions:	1. A goals details will be viewed by the client		
Normal Flow:	1. Client double-clicks on goal to be viewed 2. Interface displays goal to client		
Alternative Flows:	1. Failed to display goal <ul style="list-style-type: none"> a. Interface could not display goal (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to goal interface 		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Will want an edit, exit, assign options?		

View Completed Goals (Administrator)

Use Case ID:	68		
Use Case Name:	View Completed Goals (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary: 1. DBMS 2. Interface		
Description:	View list of completed goals		
Trigger:	Administrator clicks on view completed goals		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at goal interface		
Postconditions:	A list of completed goals will be displayed		
Normal Flow:	1. Administrator clicks on view completed goals 2. Interface displays list of all completed goal 3. Administrator clicks exit		
Alternative Flows:	1. Failed to display list of goals a. Interface cannot display goals (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to goal interface		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Formatting options to sort or filter goals		

View Completed Goals (Caregiver)

Use Case ID:	69		
Use Case Name:	View Completed Goals (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 1. DBMS 2. Interface		
Description:	View list of completed goals		
Trigger:	Caregiver clicks on view completed goals		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at goal interface		
Postconditions:	1. A list of completed goals will be displayed		
Normal Flow:	1. Caregiver clicks on view completed goals 2. Interface displays list of all completed goal		
Alternative Flows:	1. Failed to display list of goals a. Interface cannot display goals (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to goal interface		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Formatting options to sort or filter goals		

View Completed Goals (Client)

Use Case ID:	70		
Use Case Name:	View Completed Goals (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30		Last Revision Date:
Actors:	Primary Actor 1. Client Secondary Actor(s) 1. DBMS 2. Interface		
Description:	View list of completed goals		
Trigger:	Client clicks on view completed goals		
Preconditions:	1. Client must be logged into account 2. Client must be at goal interface		
Postconditions:	1. A list of completed goals will be displayed		
Normal Flow:	1. Client clicks on view completed goals 2. Interface displays list of all completed goal 3. Client clicks exit		
Alternative Flows:	1. Failed to display list of goals <ol style="list-style-type: none"> Interface cannot display goals (normal flow step 2) Interface sends error message to Administrator Interface returns to goal interface 		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Formatting options to sort or filter goals		

Remove Extinction Goal (Administrator)

Use Case ID:	71		
Use Case Name:	Remove Extinction Goal (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Remove Extinction Goal from list of assigned goal		
Trigger:	Administrator selects remove goal		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at goal interface		
Postconditions:	1. Goal will be removed from list of assigned goals		
Normal Flow:	1. Administrator selects client 2. Administrator clicks on/highlights goal 3. Administrator clicks on remove goal 4. Interface removes goal from goal list 5. Changes saved at DBMS		
Alternative Flows:	1. Failed to Save Removal of Goal (select SAVE again) a. DBMS cannot save goal (normal flow step 4) b. Interface sends error message to Administrator c. Interface asks if want to try to save again or exit d. Return to normal flow step 3 2. Failed to save removal of goal (exit) a. DBMS cannot save goal (normal flow step 4) b. Interface sends error message to Administrator c. Interface asks if want to try to save again or exit d. Exit goal creation interface.		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Do same way as assign? Drag or click		

Remove Extinction Goal (Client)

Use Case ID:	72		
Use Case Name:	Remove Extinction Goal (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Remove Extinction Goal from list of assigned goals		
Trigger:	Client select remove goal		
Preconditions:	1. Client must be logged into account 2. Client must be at goal interface		
Postconditions:	1. Goal will be removed from list of assigned goals		
Normal Flow:	1. Client clicks on//highlights goal 2. Client clicks on remove goal 3. Interface removes goal from goal list 4. Changes saved at DBMS		
Alternative Flows:	1. Failed to Save Removal of Goal (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Client Interface asks if want to try to save again or exit Return to normal flow step 3 2. Failed to save removal of goal (exit) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Client Interface asks if want to try to save again or exit Exit goal creation interface. 		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Do same way as assign? Drag or click		

Deactivate Extinction Goal (Administrator)

Use Case ID:	73		
Use Case Name:	Deactivate Extinction Goal (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Deactivate Extinction Goal from list of goal options to be assigned		
Trigger:	Administrator clicks delete/deactivate goal		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at goal interface		
Postconditions:	1. Goal will be removed from list of goals options		
Normal Flow:	1. Administrator selects client 2. Administrator selects goal 3. Administrator clicks on delete/deactivate goal 4. Administrator confirms delete/deactivate goal 5. Interface deactivate goal on list of goal options 6. Changes saved at DBMS		
Alternative Flows:	4. Failed to Save Deactivation of Goal (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Return to normal flow step 3 5. Failed to save deactivation of goal (exit) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Exit goal creation interface. 		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			

Assumptions:	
Notes and Issues:	Name of button? How are we doing interaction with diff clients

Deactivate Extinction Goal (Client)

Use Case ID:	74		
Use Case Name:	Deactivate Extinction Goal (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30		Last Revision Date:
Actors:	Primary Actor 1. Client Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Deactivate Extinction Goal from list of goal options to be assigned		
Trigger:	Client clicks delete/deactivate goal		
Preconditions:	1. Client must be logged into account 2. Client must be at goal interface		
Postconditions:	1. Goal will be removed from list of goals options		
Normal Flow:	1. Client clicks on//highlights goal 2. Client clicks on delete/deactivate goal 3. Client confirms delete/deactivate goal 4. Interface deactivate goal on list of goal options 5. Changes saved at DBMS		
Alternative Flows:	1. Failed to Save Deactivation of Goal (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Client Interface asks if want to try to save again or exit Return to normal flow step 3 2. Failed to save deactivation of goal (exit) <ol style="list-style-type: none"> DBMS cannot save goal (normal flow step 4) Interface sends error message to Client Interface asks if want to try to save again or exit Exit goal creation interface. 		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			

Notes and Issues:	1.Name of button?
--------------------------	-------------------

View Goal Details

Use Case ID:	75		
Use Case Name:	View Goal Details		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 2. DBMS 3. Interface		
Description:	See the details of a goals creation to be able to audit goals		
Trigger:	Administrator clicks delete/deactivate goal		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at goal interface		
Postconditions:	Goal will be removed from list of goals options		
Normal Flow:	1. Administrator selects client 2. Administrator clicks Goal Audit 3. Interface displays background goal information 4. Administrator clicks exit		
Alternative Flows:	1. Failed to display audit of goal a. Interface cannot display goals (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to goal interface		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:	We saved details of goal creation, like who created, when, use? etc.		
Notes and Issues:	Name of button?		

Create Incident (Administrator)

Use Case ID:	76		
Use Case Name:	Create Incident (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Create an incident		
Trigger:	Administrator clicks create incident		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at incident interface		
Postconditions:	An incident will be created		
Normal Flow:	1. Administrator clicks create incident 2. Administrator enters information 3. Administrator clicks save 4. Interface saves incident 5. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Administrator corrects data. Return to normal flow step 2. 2. Failed to Save Incident (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Return to normal flow step 3 3. Failed to save incident(exit) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Exit incident creation interface. 		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		

Special Requirements:	
Assumptions:	
Notes and Issues:	Name of button How are we moving between interfaces

Create Incident (Caregiver)

Use Case ID:	77		
Use Case Name:	Create Incident (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Create an incident		
Trigger:	Caregiver clicks create incident		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at incident interface		
Postconditions:	An incident will be created		
Normal Flow:	1. Caregiver clicks create incident 2. Caregiver enters information 3. Caregiver clicks save 4. Interface saves incident 5. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Caregiver corrects data. Return to normal flow step 2. 2. Failed to Save Incident (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Caregiver Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save incident (exit)</p> <ul style="list-style-type: none"> a. DBMS cannot save incident (normal flow step 4) b. Interface sends error message to Caregiver c. Interface asks if want to try to save again or exit d. Exit incident creation interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	Name of button? interface interactions

Create Incident (Client)

Use Case ID:	78		
Use Case Name:	Create Incident (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Create an incident		
Trigger:	Client clicks create incident		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident will be created		
Normal Flow:	1. Client clicks create incident 2. Client enters information 3. Client clicks save 4. Interface saves incident 5. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Client corrects data. Return to normal flow step 2. 2. Failed to Save Incident (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Client Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save incident (exit)</p> <p>3. DBMS cannot save incident (normal flow step 4)</p> <ul style="list-style-type: none"> b. Interface sends error message to Client c. Interface asks if want to try to save again or exit d. Exit incident creation interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	Name of button?

Edit Incident (Administrator)

Use Case ID:	79		
Use Case Name:	Edit Incident (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Create an incident		
Trigger:	Administrator clicks edit incident		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at incident interface		
Postconditions:	An incident will be edited		
Normal Flow:	3. Administrator clicks edit incident 4. Administrator edits information 5. Administrator clicks save 6. Interface saves incident 7. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Administrator corrects data. Return to normal flow step 2. 2. Failed to Save Incident (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save incident (exit)</p> <p>4. DBMS cannot save incident (normal flow step 4)</p> <ul style="list-style-type: none"> b. Interface sends error message to Administrator c. Interface asks if want to try to save again or exit d. Exit incident edit interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	Name of button?

Edit Incident (Client)

Use Case ID:	80		
Use Case Name:	Edit Incident (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Edit an incident		
Trigger:	Client clicks edit incident		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident will be edited		
Normal Flow:	1. Client clicks edit incident 2. Client edits information 3. Client clicks save 4. Interface saves incident 5. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Client corrects data. Return to normal flow step 2. 2. Failed to Save Incident (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Client Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save incident (exit)</p> <p>5. DBMS cannot save incident (normal flow step 4)</p> <ul style="list-style-type: none"> b. Interface sends error message to Client c. Interface asks if want to try to save again or exit d. Exit incident edit interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	Name of button?

Edit Incident (Caregiver)

Use Case ID:	81		
Use Case Name:	Edit Incident (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30		Last Revision Date:
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Edit an incident		
Trigger:	Caregiver clicks edit incident		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at incident interface		
Postconditions:	An incident will be edited		
Normal Flow:	1. Caregiver clicks edit incident 2. Caregiver edits information 3. Caregiver clicks save 4. Interface saves incident 5. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Caregiver corrects data. Return to normal flow step 2. 2. Failed to Save Incident (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Caregiver Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save incident (exit)</p> <p>6. DBMS cannot save incident (normal flow step 4)</p> <ul style="list-style-type: none"> b. Interface sends error message to Caregiver c. Interface asks if want to try to save again or exit d. Exit incident edit interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	Name of button?

Assign Incident (Administrator)

Use Case ID:	82		
Use Case Name:	Assign Incident (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Incident will be assigned to a client		
Trigger:	Administrator clicks assign incident		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at incident interface		
Postconditions:	An incident will be assigned to a client		
Normal Flow:	1. Administrator selects client 2. Administrator selects incident 3. Administrator assigns an incident to client 4. Administrator clicks save 5. Interface saves incident 6. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Administrator corrects data. Return to normal flow step 2. 2. Failed to save incident assignment (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Administrator Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save incident assignment (exit)</p> <p>7. DBMS cannot save incident (normal flow step 4)</p> <ul style="list-style-type: none"> b. Interface sends error message to Administrator c. Interface asks if want to try to save again or exit d. Exit assign incident interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	That we have a way to select a client while in interfaces
Notes and Issues:	Name of button? Want to do from client or incident interface or both?

Assign Incident (Client)

Use Case ID:	83		
Use Case Name:	Assign Incident (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Incident will be assigned to client		
Trigger:	Client clicks assign incident		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident will be assigned to client		
Normal Flow:	1. Client clicks assign incident 2. Client assigns an incident 3. Client clicks save 4. Interface saves incident 5. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Client corrects data. Return to normal flow step 2. 2. Failed to save incident assignment (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Client Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save incident assignment (exit)</p> <p>8. DBMS cannot save incident (normal flow step 4)</p> <ul style="list-style-type: none"> b. Interface sends error message to Client c. Interface asks if want to try to save again or exit d. Exit assign incident interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	
Notes and Issues:	Name of button?

Assign Incident (Caregiver)

Use Case ID:	84		
Use Case Name:	Assign Incident (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Incident will be assigned to a client		
Trigger:	Caregiver clicks assign incident		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at incident interface		
Postconditions:	An incident will be assigned to a client		
Normal Flow:	1. Caregiver clicks assign incident 2. Caregiver selects client 3. Caregiver assigns an incident to client 4. Caregiver clicks save 5. Interface saves incident 6. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Caregiver corrects data. Return to normal flow step 2. 2. Failed to save incident assignment (select SAVE again) <ol style="list-style-type: none"> DBMS cannot save incident (normal flow step 4) Interface sends error message to Caregiver Interface asks if want to try to save again or exit Return to normal flow step 3 		

	<p>3. Failed to save incident assignment (exit)</p> <p>9. DBMS cannot save incident (normal flow step 4)</p> <ul style="list-style-type: none"> b. Interface sends error message to Caregiver c. Interface asks if want to try to save again or exit d. Exit assign incident interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	That we have a way to select a client while in interfaces
Notes and Issues:	7. Name of button?

View Incident (Administrator)

Use Case ID:	85		
Use Case Name:	View Incident (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Incident details will be displayed		
Trigger:	Administrator double click on incident OR Administrator highlights incident and clicks view		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at incident interface		
Postconditions:	An incident will be displayed		
Normal Flow:	1. Administrator highlights incident 2. Administrator selects view incident 3. Interface displays incident 4. Administrator clicks exit		
Alternative Flows:	1. Failed to display incident a. Interface cannot display incident (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to incident interface		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Name of button?		

View Incident (Client)

Use Case ID:	86		
Use Case Name:	View Incident (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Incident details will be displayed		
Trigger:	1. Client double-clicks on incident 2. Client highlights incident and clicks view		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident will be displayed		
Normal Flow:	1. Client highlights incident 2. Client selects view incident 3. Interface displays incident 4. Client clicks exit		
Alternative Flows:	1. Failed to display incident a. Interface cannot display incident (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to incident interface		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Name of button?		

View Incident (Caregiver)

Use Case ID:	87		
Use Case Name:	View Incident (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Incident details will be displayed		
Trigger:	1. Caregiver double clicks on incident 2. Caregiver highlights incident and clicks view		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at incident interface		
Postconditions:	An incident will be displayed		
Normal Flow:	1. Caregiver highlights incident 2. Caregiver selects view incident 3. Interface displays incident 4. Caregiver clicks exit		
Alternative Flows:	1. Failed to display incident a. Interface cannot display incident (normal flow step 2) b. Interface sends error message to Administrator c. Interface returns to incident interface		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:			
Notes and Issues:	Name of button? Also send error message to administrator.		

Track Incidents Over Time (Administrator)

Use Case ID:	88		
Use Case Name:	Track Incidents Over Time (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display an incidents occurrence over time by a client		
Trigger:	Administrator clicks View Over Time		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at view incident interface		
Postconditions:	An incident will be displayed		
Normal Flow:	1. Administrator selects incident to be viewed 2. Administrator selects client 3. Administrator clicks View Over Time 4. Interface displays incident over time 5. Administrator clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to Administrator Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Administrator corrects data Return to normal flow step1 or 2. 		
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too?		

Track Incidents Over Time (Client)

Use Case ID:	89		
Use Case Name:	Track Incidents Over Time (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30		Last Revision Date:
Actors:	Primary Actor 1. Client Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Interface will display an incidents occurrence over time		
Trigger:	Client clicks View Over Time		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident will be displayed		
Normal Flow:	1. Client selects incident to be viewed 2. Client clicks View Over Time 3. Interface displays incident over time 4. Client clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to client Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Client corrects data Return to normal flow step1 or 2. 		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too?		

Track Incidents Over Time (Caregiver)

Use Case ID:	90		
Use Case Name:	Track Incidents Over Time (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display an incidents occurrence over time by a client		
Trigger:	Caregiver clicks View Over Time		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at view incident interface		
Postconditions:	An incident will be displayed		
Normal Flow:	1. Caregiver selects client 2. Caregiver selects incident to be viewed 3. Caregiver clicks View Over Time 4. Interface displays incident over time 5. Caregiver clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to caregiver Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Caregiver corrects data Return to normal flow step1 or 2. 		
Includes:			
Frequency of Use:	On Demand		
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection, Notify administrator too?		

See Report of Incidents Based on Date (Administrator)

Use Case ID:	91		
Use Case Name:	See Report of Incidents Based on Date (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display an incident occurrence by selected date		
Trigger:	Administrator clicks View Incident by Date		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at incident interface		
Postconditions:	An incident will be displayed by date		
Normal Flow:	1. Administrator selects client 2. Administrator selects incident to be viewed 3. Administrator selects date to be viewed 4. Administrator clicks View Incident by Date 5. Interface displays incident by date 6. Administrator clicks exit		
Alternative Flows:	1. Failed to display results a. Interface cannot display incident (normal flow step 2) b. Interface sends error message to administrator c. Interface returns to incident interface 2. Invalid Data Entry a. Invalid data entered (normal flow step1 or 2) b. Interface sends error message to screen c. Administrator corrects data d. Return to normal flow step1 or 2.		
Exceptions:			
Frequency of Use:	On Demand		
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client		

See Report of Incidents Based on Date (Client)

Use Case ID:	92		
Use Case Name:	See Report of Incidents Based on Date (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30		Last Revision Date:
Actors:	Primary Actor 1. Client Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display an incident occurrence by selected date		
Trigger:	Client clicks View Incident by Date		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident will be displayed by date		
Normal Flow:	1. Client selects incident to be viewed 2. Client selects date to be viewed 3. Client clicks View Incident by Date 4. Interface displays incident by date 5. Client clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to client Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Client corrects data Return to normal flow step1 or 2. 		
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client		

See Report of Incidents Based on Date (Caregiver)

Use Case ID:	93		
Use Case Name:	See Report of Incidents Based on Date (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display an incident occurrence by selected date		
Trigger:	Caregiver clicks View Incident by Date		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at incident interface		
Postconditions:	An incident will be displayed by date		
Normal Flow:	1. Caregiver selects incident to be viewed 2. Caregiver selects client 3. Caregiver selects date to be viewed 4. Caregiver clicks View Incident by Date 5. Interface displays incident by date 6. Caregiver clicks exit		
Alternative Flows:	1. Failed to display results a. Interface cannot display incident (normal flow step 2) b. Interface sends error message to caregiver c. Interface returns to incident interface 2. Invalid Data Entry a. Invalid data entered (normal flow step1 or 2) b. Interface sends error message to screen c. Caregiver corrects data d. Return to normal flow step1 or 2.		
Frequency of Use:	On Demand		
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client		

See Dates Based on Type/Name of Incident (Administrator)

Use Case ID:	94		
Use Case Name:	See Dates Based on Type/Name of Incident (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Interface will display the dates of occurrence by selected type/name of incident		
Trigger:	Administrator clicks View Dates by Incident		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at incident interface		
Postconditions:	An incident will be displayed by date		
Normal Flow:	1. Administrator selects client 2. Administrator selects incident to be viewed 3. Administrator selects date to be viewed 4. Administrator clicks View Dates by Incident 5. Interface displays dates of incident 6. Administrator clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to administrator Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Administrator corrects data Return to normal flow step1 or 2. 		
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client		

See Dates Based on Type/Name of Incident (Client)

Use Case ID:	95		
Use Case Name:	See Dates Based on Type/Name of Incident (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Interface will display the dates of occurrence by selected type/name of incident		
Trigger:	Client clicks View Dates by Incident		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident will be displayed by date		
Normal Flow:	1. Client selects incident to be viewed 2. Client selects date to be viewed 3. Client clicks View Dates by Incident 4. Interface displays dates of incident 5. Client clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to client Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Client corrects data Return to normal flow step1 or 2. 		
Frequency of Use:	On Demand		
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client		

See Dates Based on Type/Name of Incident (Caregiver)

Use Case ID:	96		
Use Case Name:	See Dates Based on Type/Name of Incident (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display the dates of occurrence by selected type/name of incident		
Trigger:	Caregiver clicks View Dates by Incident		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at incident interface		
Postconditions:	An incident will be displayed by date		
Normal Flow:	1. Caregiver selects incident to be viewed 2. Caregiver selects client 3. Caregiver selects date to be viewed 4. Caregiver clicks View Dates by Incident 5. Interface displays dates of incident 6. Caregiver clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to caregiver Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Caregiver corrects data Return to normal flow step1 or 2. 		
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client		

See Details of Specific Incident (Administrator)

Use Case ID:	97		
Use Case Name:	See Details of Specific Incident (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display the details of specific incident		
Trigger:	Administrator clicks View Details of Incident		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at incident interface		
Postconditions:	An incident details will be displayed		
Normal Flow:	1. Administrator selects client 2. Administrator selects specific client incident 3. Administrator clicks View Details 4. Interface displays details of incident 5. Administrator clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to administrator Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Administrator corrects data Return to normal flow step1 or 2. 		
Frequency of Use:	On Demand		
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client		

See Details of Specific Incident (Client)

Use Case ID:	98		
Use Case Name:	See Details of Specific Incident (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display the details of specific incident		
Trigger:	Client clicks View Details of Incident		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident details will be displayed		
Normal Flow:	1. Client selects specific incident 2. Client clicks View Details 3. Interface displays details of incident 4. Client clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to client Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Client corrects data Return to normal flow step1 or 2. 		
Exceptions:			
Includes:			
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:	That we have a way to select a client while in interfaces		

Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client
--------------------------	--

See Details of Specific Incident (Caregiver)

Use Case ID:	UC-11.8.3		
Use Case Name:	See Details of Specific Incident (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Interface will display the details of specific incident		
Trigger:	Caregiver clicks View Details of Incident		
Preconditions:	1. Caregiver must be logged into account 2. Caregiver must be at incident interface		
Postconditions:	An incident details will be displayed		
Normal Flow:	1. Caregiver selects client 2. Caregiver selects specific client incident 3. Caregiver clicks View Details 4. Interface displays details of incident 5. Caregiver clicks exit		
Alternative Flows:	1. Failed to display results <ol style="list-style-type: none"> Interface cannot display incident (normal flow step 2) Interface sends error message to client Interface returns to incident interface 2. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step1 or 2) Interface sends error message to screen Client corrects data Return to normal flow step1 or 2. 		
Frequency of Use:	On Demand		
Special Requirements:			
Assumptions:	That we have a way to select a client while in interfaces		
Notes and Issues:	Name of button? Order of selection; Notify administrator too? Specific date or range date; One specific type of incident or all incidents by a specific client		

Record Incident (Administrator)

Use Case ID:	100		
Use Case Name:	Record Incident (Administrator)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Administrator Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Record an incident		
Trigger:	Administrator clicks Record Incident		
Preconditions:	1. Administrator must be logged into account 2. Administrator must be at incident interface		
Postconditions:	An incident will be recorded		
Normal Flow:	1. Administrator selects client 2. Administrator clicks record incident 3. Interface displays list of incidents for client 4. Administrator clicks incident 5. Administrator enters incident details 6. Administrator clicks save 7. Interface saves incident 8. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Administrator corrects data. Return to normal flow step 2. 2. Need new incident <ol style="list-style-type: none"> Administrator selects Create Incident (normal flow step 3) Administrator enters Incident details Administrator clicks save Interface saves incident Changes saved at DBMS 		

	<p>f. Interface returns to Record Incident interface (normal flow step 4)</p> <p>2. Failed to Save Incident (select SAVE again)</p> <ul style="list-style-type: none"> a. DBMS cannot save incident (normal flow step 4) b. Interface sends error message to Administrator c. Interface asks if want to try to save again or exit d. Return to normal flow step 3 <p>3. Failed to save incident (exit)</p> <ul style="list-style-type: none"> 10. DBMS cannot save incident (normal flow step 4) b. Interface sends error message to Administrator c. Interface asks if want to try to save again or exit d. Exit incident creation interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	Will auto input date/time, but can be changed in case recorded later
Notes and Issues:	Name of button?

Record Incident (Client)

Use Case ID:	101		
Use Case Name:	Record Incident (Client)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Client Secondary Actor(s) 1. DBMS 2. Interface		
Description:	Record an incident		
Trigger:	Client clicks Record Incident		
Preconditions:	1. Client must be logged into account 2. Client must be at incident interface		
Postconditions:	An incident will be recorded		
Normal Flow:	1. Client clicks record incident 2. Interface displays list of incidents 3. Client clicks incident 4. Client enters incident details 5. Client clicks save 6. Interface saves incident 7. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Client corrects data. Return to normal flow step 2. 2. Need new incident <ol style="list-style-type: none"> Client selects Create Incident (normal flow step 3) Client enters Incident details Client clicks save Interface saves incident Changes saved at DBMS 		

	<p>f. Interface returns to Record Incident interface (normal flow step 4)</p> <p>2. Failed to Save Incident (select SAVE again)</p> <ul style="list-style-type: none"> a. DBMS cannot save incident (normal flow step 4) b. Interface sends error message to Client c. Interface asks if want to try to save again or exit d. Return to normal flow step 3 <p>3. Failed to save incident (exit)</p> <ul style="list-style-type: none"> a. DBMS cannot save incident (normal flow step 4) b. Interface sends error message to Client c. Interface asks if want to try to save again or exit d. Exit incident creation interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	Will auto input date/time, but can be changed in case recorded later
Notes and Issues:	Name of button?

Record Incident (Caregiver)

Use Case ID:	102		
Use Case Name:	Record Incident (Caregiver)		
Created By:	Becky	Last Updated By:	
Date Created:	2020-10-30	Last Revision Date:	
Actors:	Primary Actor 1. Caregiver Secondary Actor(s) 2. DBMS 3. Interface		
Description:	Record an incident		
Trigger:	Caregiver clicks Record Incident		
Preconditions:	1. Record must be logged into account 2. Record must be at incident interface		
Postconditions:	An incident will be recorded		
Normal Flow:	1. Caregiver selects client 2. Caregiver clicks record incident 3. Interface displays list of incidents for client 4. Caregiver clicks incident 5. Caregiver enters incident details 6. Caregiver clicks save 7. Interface saves incident 8. Changes saved at DBMS		
Alternative Flows:	1. Invalid Data Entry <ol style="list-style-type: none"> Invalid data entered (normal flow step 2) Interface sends error message to screen Caregiver corrects data. Return to normal flow step 2. 2. Need new incident <ol style="list-style-type: none"> Caregiver selects Create Incident (normal flow step 3) Caregiver enters Incident details Caregiver clicks save Interface saves incident 		

	<p>e. Changes saved at DBMS</p> <p>f. Interface returns to Record Incident interface (normal flow step 4)</p> <p>2. Failed to Save Incident (select SAVE again)</p> <ul style="list-style-type: none"> a. DBMS cannot save incident (normal flow step 4) b. Interface sends error message to Caregiver c. Interface asks if want to try to save again or exit d. Return to normal flow step 3 <p>3. Failed to save incident (exit)</p> <ul style="list-style-type: none"> 3. DBMS cannot save incident (normal flow step 4) b. Interface sends error message to Caregiver c. Interface asks if want to try to save again or exit d. Exit incident creation interface.
Exceptions:	
Includes:	
Frequency of Use:	On Demand
Special Requirements:	
Assumptions:	Will auto input date/time, but can be changed in case recorded later
Notes and Issues:	Name of button?

Create Journal

Use Case ID:	111	
Use Case Name:	Create Journal	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User creates a Learning Journal	
Trigger:	User wants to create a Journal	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.)	
Postconditions:	1. User has a created Journal	
Normal Flow:	<ol style="list-style-type: none"> 1. User selects “Create New Journal” 2. System Prompts User for Name of Journal 3. User types Name of Journal 4. System prompts user for Topic of Journal 5. User enters Topic of Journal 6. System prompts user to Validate Info 7. User Validates info 	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	6a. In step 6 of the normal flow, if the User wants to edit info 1. Use case loops back to step 2	
Exceptions:		
Includes:		
Frequency of Use:	Once a month per User	
Special Requirements:	N/A	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Create Journal Entry

Use Case ID:	112		
Use Case Name:	Create Journal Entry		
Created By:	Nathaniel Webber	Last Updated By:	
Date Created:	09/24/20210	Last Revision Date:	
Actors:	Adult with Physical Disability		
Description:	User creates a Journal Entry		
Trigger:	User wants to create a Journal Entry		
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.)		
Postconditions:	1. User has a created Journal Entry		
Normal Flow:	<ol style="list-style-type: none"> 1. User selects "My Journals" 2. System Prompts User for Name of Journal 3. User selects Name of Journal 4. System prompts user for Description of Journal Entry 5. User enters Description of Journal Entry 6. System prompts user to Validate Info 7. User Validates info 8. System posts Journal Entry 		
Alternative Flows:			
[Alternative Flow 1 – Not in Network]	<p>6a. In step 6 of the normal flow, if the User wants to edit info</p> <ol style="list-style-type: none"> 1. Use case loops back to step 2 		
Exceptions:			
Includes:			
Frequency of Use:	Once a month per User		
Special Requirements:	[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]		
Assumptions:	User understands English and has basic computer competency		
Notes and Issues:	<ol style="list-style-type: none"> 1. 		

Create Profile

Use Case ID:	113	
Use Case Name:	Create Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User creates a Learning Goal	
Trigger:	User wants to create a Learning Goal	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.)	
Postconditions:	1. User has a created Learning Goal	
Normal Flow:	<ol style="list-style-type: none"> 1. User selects “Create Learning Goal” 2. System Prompts User for Name of Goal 3. User types Name of Goal 4. System prompts User for Type of Goal 5. User enters Type of Goal 6. System prompts user for Description of Goal 7. User enters Description of Goal 8. System prompts User to set Reminder for Goal Progress 9. User sets Reminder 10. System prompts user to Validate Info 11. User Validates info 	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	<p>10a. In step 10 of the normal flow, if the User wants to edit info</p> <ol style="list-style-type: none"> 1. Use case loops back to step 2 	
Frequency of Use:	Once a month per User	
Special Requirements:	[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	<ol style="list-style-type: none"> 1. 	

Create Profile

Use Case ID:	114	
Use Case Name:	Create Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/18/2020	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User creates a profile	
Trigger:	User wants to create a profile	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.)	
Postconditions:	1. User has a created profile	
Normal Flow:	<ol style="list-style-type: none"> 1. User selects “Create Profile” 2. System Prompts User for Name 3. User types Name 4. System prompts User for DOB 5. User enters DOB 6. System prompts user for Email 7. User enters Email 8. System prompts User to Validate info 9. User validates info 	
Alternative Flows: [Alternative Flow 1 – Not in Network]	<p>6a. In step 6 of the normal flow, if the User does not have an Email</p> <ol style="list-style-type: none"> 1. System will prompt User to either create an Email or enter a valid Email 2. Customer accepts 3. Use Case resumes on step 7 	
Frequency of Use:	Once per user	
Special Requirements:	[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Delete Journal Entry

Use Case ID:	115		
Use Case Name:	Delete Journal Entry		
Created By:	Nathaniel Webber	Last Updated By:	
Date Created:	09/24/20210	Last Revision Date:	
Actors:	Adult with Physical Disability		
Description:	User Deletes Journal Entry		
Trigger:	User wants to Delete a Journal Entry		
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have a Journal Entry		
Postconditions:			
Normal Flow:	1. User Selects “Delete Journal Entry” 2. System displays Journals 3. User Selects the Journal containing the Entry to Delete 4. System displays Journal Entries 5. User Selects Journal Entry to delete 6. System prompts user to confirm choice 7. User Validates Choice 8. System Deactivates User Account		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:			
Includes:			
Frequency of Use:	Once a year per user		
Special Requirements:	User must have a Profile		
Assumptions:	User understands English and has basic computer competency		
Notes and Issues:	1.		

Delete Profile

Use Case ID:	116		
Use Case Name:	Delete Profile		
Created By:	Nathaniel Webber	Last Updated By:	
Date Created:	09/24/20210	Last Revision Date:	
Actors:	Adult with Physical Disability		
Description:	User Deletes Learning Goal		
Trigger:	User wants to Delete a Learning Goal		
Preconditions:	<ol style="list-style-type: none">1. User must have a device with the program on it (PC, Tablet, etc.)2. User must have a Learning Goal		
Postconditions:			
Normal Flow:	<ol style="list-style-type: none">1. User Selects “Delete Learning Goal”2. System displays Learning Goals3. User Selects Learning Goal to delete4. System prompts user to confirm choice5. User Validates Choice6. System Deactivates User Account		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:			
Includes:			
Frequency of Use:	Once a year per user		
Special Requirements:	User must have a Profile		
Assumptions:	User understands English and has basic computer competency		
Notes and Issues:	1.		

Delete Profile

Use Case ID:	117	
Use Case Name:	Delete Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User Deletes Profile	
Trigger:	User wants to Delete Profile	
Preconditions:	<ol style="list-style-type: none">1. User must have a device with the program on it (PC, Tablet, etc.)2. User must have a Profile	
Postconditions:		
Normal Flow:	<ol style="list-style-type: none">1. User Selects “Delete Profile”2. System prompts user to confirm choice3. User Validates Choice4. System Deactivates User Account	
Alternative Flows: [Alternative Flow 1 – Not in Network]		
Exceptions:		
Includes:		
Frequency of Use:	Once per User	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Retrieve Journal Entries

Use Case ID:	118	
Use Case Name:	Retrieve Journal Entries	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User Retrieves Journal Entries	
Trigger:	User wants to View their various Journal Entries	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Profile	
Postconditions:		
Normal Flow:	1. User Selects "View Journals" 2. System Lists all Journals 3. User Selects Journal they wish to view 4. System displays all Journal Entries in Journal	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	1a. In step 1 of the normal flow, if the User has no Journals 1. System prompts User to create a new Journal	
Exceptions:		
Includes:		
Frequency of Use:	Once a day	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Retrieve Profile

Use Case ID:	119	
Use Case Name:	Retrieve Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User Retrieves Learning Goal	
Trigger:	User wants to View their various Learning Goals	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Profile	
Postconditions:	1. User has a Logged into their profile	
Normal Flow:	1. User Selects "View Learning Goals" 2. System Lists all Learning Goals	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	1a. In step 1 of the normal flow, if the User has no Learning Goals 1. System prompts User to create a new Learning Goal	
Exceptions:		
Includes:		
Frequency of Use:	Once a day	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Retrieve Profile

Use Case ID:	120	
Use Case Name:	Retrieve Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User Retrieves Profile	
Trigger:	User wants to create a Log in to their Profile	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Profile	
Postconditions:	1. User has a Logged into their profile	
Normal Flow:	1. System Prompts User for Email 2. User Enters Email 3. System prompts user for Password 4. User enters Password 5. User Selects "Log in" 6. System Validates User 7. System Logs in User	
Alternative Flows:	[Alternative Flow 1 – Not in Network] 3a. In step 3 of the normal flow, if the User enters invalid Email or Password <ul style="list-style-type: none"> 1. System prompts User saying Email or Password was not correct 2. System prompts User to Re-enter Email and Password 3. System prompts user to reset password if they want to, or create a new profile 	
Exceptions:		
Includes:		
Frequency of Use:	Once a day	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Update Journal Entry

Use Case ID:	121	
Use Case Name:	Update Journal Entry	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User Updates Journal Entry	
Trigger:	User wants to Update their Journal Entry	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Learning Goal	
Postconditions:		
Normal Flow:	1. User Selects "Update Journal Entry" 2. System prompts user to select a Journal to edit 3. User Selects the Journal to edit 4. System prompts user to select Journal Entry to edit 5. User selects Journal entry to edit 6. System prompts user to edit Info 7. User Edits info 8. System prompts user to confirm info 9. User Validates info 10. System saves changes	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	9a. In step 9 of the normal flow, if the User wants to re-edit info 1. Return to Normal Flow Step 5	
Exceptions:		
Includes:		
Frequency of Use:	Once a day	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Update Profile

Use Case ID:	122	
Use Case Name:	Update Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User Updates Learning Goal	
Trigger:	User wants to Update their Learning Goal	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Learning Goal	
Postconditions:		
Normal Flow:	1. User Selects "Update Learning Goal" 2. System prompts user to select a goal to edit 3. User Selects the Learning goal to edit 4. System prompts user to edit Info 5. User Edits info 6. System prompts user to confirm info 7. User Validates info 8. System saves changes	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	7a. In step 7 of the normal flow, if the User wants to re-edit info 1. Return to Normal Flow Step 4	
Exceptions:		
Includes:		
Frequency of Use:	Once a day	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Update Profile

Use Case ID:	123		
Use Case Name:	Update Profile		
Created By:	Nathaniel Webber	Last Updated By:	
Date Created:	09/24/20210	Last Revision Date:	
Actors:	Adult with Physical Disability		
Description:	User Updates Profile		
Trigger:	User wants to Update their Profile		
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Profile		
Postconditions:			
Normal Flow:	1. User Selects "Update Profile" 2. System displays Edit Profile Screen 3. User Edits info 4. System prompts user to confirm info 5. User validates info 6. System saves changes		
Alternative Flows:			
[Alternative Flow 1 – Not in Network]	4a. In step 4 of the normal flow, if the User wants to re-edit info 1. Return to Normal Flow Step 2		
Exceptions:			
Includes:			
Frequency of Use:	Once a day		
Special Requirements:	User must have a Profile		
Assumptions:	User understands English and has basic computer competency		
Notes and Issues:	1.		

Create Journal

Use Case ID:	124	
Use Case Name:	Create Journal	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Elder with Cognitive Disability	
Description:	User creates a Learning Journal	
Trigger:	User wants to create a Journal	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.)	
Postconditions:	1. User has a created Journal	
Normal Flow:	<ol style="list-style-type: none"> 1. User selects “Create New Journal” 2. System Prompts User for Name of Journal 3. User types Name of Journal 4. System prompts user for Topic of Journal 5. User enters Topic of Journal 6. System prompts user to Validate Info 7. User Validates info 	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	6a. In step 6 of the normal flow, if the User wants to edit info 1. Use case loops back to step 2	
Exceptions:		
Includes:		
Frequency of Use:	Once a month per User	
Special Requirements:	[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Create Journal Entry

Use Case ID:	125	
Use Case Name:	Create Journal Entry	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Elder with Cognitive Disability	
Description:	User creates a Journal Entry	
Trigger:	User wants to create a Journal Entry	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.)	
Postconditions:	1. User has a created Journal Entry	
Normal Flow:	<ol style="list-style-type: none"> 1. User selects "My Journals" 2. System Prompts User for Name of Journal 3. User selects Name of Journal 4. System prompts user for Description of Journal Entry 5. User enters Description of Journal Entry 6. System prompts user to Validate Info 7. User Validates info 8. System Posts Journal Entry 	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	<p>6a. In step 6 of the normal flow, if the User wants to edit info</p> <ol style="list-style-type: none"> 1. Use case loops back to step 2 	
Exceptions:		
Frequency of Use:	Once a month per User	
Special Requirements:	[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Create Profile

Use Case ID:	126	
Use Case Name:	Create Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Elder with Cognitive Disability	
Description:	User creates a Learning Goal	
Trigger:	User wants to create a Learning Goal	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.)	
Postconditions:	1. User has a created Learning Goal	
Normal Flow:	2. User selects “Create Learning Goal” 3. System Prompts User for Name of Goal 4. User types Name of Goal 5. System prompts User for Type of Goal 6. User enters Type of Goal 7. System prompts user for Description of Goal 7. User enters Description of Goal 8. System prompts User to set Reminder for Goal Progress 9. User sets Reminder 10. System prompts user to Validate Info 11. User Validates info	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	10a. In step 10 of the normal flow, if the User wants to edit info 1. Use case loops back to step 2	
Exceptions:		
Frequency of Use:	Once a month per User	
Special Requirements:	[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:		

Create Profile

Use Case ID:	127	
Use Case Name:	Create Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/2020	Last Revision Date:
Actors:	Elder with Cognitive Disability	
Description:	User creates a profile	
Trigger:	User wants to create a profile	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.)	
Postconditions:	1. User has a created profile	
Normal Flow:	<ol style="list-style-type: none"> 1. User selects “Create Profile” 2. System Prompts User for Name 3. User types Name 4. System prompts User for DOB 5. User enters DOB 6. System prompts user for Email 7. User enters Email 8. System prompts User to Validate info 9. User validates info 	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	<p>6a. In step 6 of the normal flow, if the User does not have an Email</p> <ol style="list-style-type: none"> 1. System will prompt User to either create an Email or enter a valid Email 2. Customer accepts 3. Use Case resumes on step 7 	
Exceptions:		
Frequency of Use:	Once per user	
Special Requirements:	[Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.]	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:		

Delete Journal Entry

Use Case ID:	128	
Use Case Name:	Delete Journal Entry	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Elder with Cognitive Disability	
Description:	User Deletes Journal Entry	
Trigger:	User wants to Delete a Journal Entry	
Preconditions:	<ol style="list-style-type: none">1. User must have a device with the program on it (PC, Tablet, etc.)2. User must have a Journal Entry	
Postconditions:		
Normal Flow:	<ol style="list-style-type: none">1. User Selects “Delete Journal Entry”2. System displays Journals3. User Selects the Journal containing the Entry to Delete4. System displays Journal Entries5. User Selects Journal Entry to delete6. System prompts user to confirm choice7. User Validates Choice8. System Deactivates User Account	
Alternative Flows: [Alternative Flow 1 – Not in Network]		
Exceptions:		
Includes:		
Frequency of Use:	Once a year per user	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	<ol style="list-style-type: none">1.	

Delete Learning Goals

Use Case ID:	129		
Use Case Name:	Delete Learning Goals		
Created By:	Nathaniel Webber	Last Updated By:	
Date Created:	09/24/20210	Last Revision Date:	
Actors:	Elder with Cognitive Disability		
Description:	User Deletes Learning Goal		
Trigger:	User wants to Delete a Learning Goal		
Preconditions:	<ol style="list-style-type: none">1. User must have a device with the program on it (PC, Tablet, etc.)2. User must have a Learning Goal		
Postconditions:			
Normal Flow:	<ol style="list-style-type: none">1. User Selects “Delete Learning Goal”2. System displays Learning Goals3. User Selects Learning Goal to delete4. System prompts user to confirm choice5. User Validates Choice6. System Deactivates User Account		
Alternative Flows: [Alternative Flow 1 – Not in Network]			
Exceptions:			
Includes:			
Frequency of Use:	Once a year per user		
Special Requirements:	User must have a Profile		
Assumptions:	User understands English and has basic computer competency		
Notes and Issues:	1.		

Delete Profile

Use Case ID:	130	
Use Case Name:	Delete Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Elder with Cognitive Disability	
Description:	User Deletes Profile	
Trigger:	User wants to Delete Profile	
Preconditions:	<ol style="list-style-type: none">1. User must have a device with the program on it (PC, Tablet, etc.)2. User must have a Profile	
Postconditions:		
Normal Flow:	<ol style="list-style-type: none">1. User Selects “Delete Profile”2. System prompts user to confirm choice3. User Validates Choice4. System Deactivates User Account	
Alternative Flows: [Alternative Flow 1 – Not in Network]		
Exceptions:		
Includes:		
Frequency of Use:	Once per User	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:		

Retrieve Journal Entries

Use Case ID:	131	
Use Case Name:	Retrieve Journal Entries	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Elder with Cognitive Disability	
Description:	User Retrieves Journal Entries	
Trigger:	User wants to View their various Journal Entries	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Profile	
Postconditions:		
Normal Flow:	1. User Selects "View Journals" 2. System Lists all Journals 3. User Selects Journal they wish to view 4. System displays all Journal Entries in Journal	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	1a. In step 1 of the normal flow, if the User has no Journals 1. System prompts User to create a new Journal	
Exceptions:		
Includes:		
Frequency of Use:	Once a day	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:		

Retrieve Profile

Use Case ID:	132	
Use Case Name:	Retrieve Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Elder with Cognitive Disability	
Description:	User Retrieves Learning Goal	
Trigger:	User wants to View their various Learning Goals	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Profile	
Postconditions:	1. User has a Logged into their profile	
Normal Flow:	1. User Selects "View Learning Goals" 2. System Lists all Learning Goals	
Alternative Flows:		
[Alternative Flow 1 – Not in Network]	1a. In step 1 of the normal flow, if the User has no Learning Goals 1. System prompts User to create a new Learning Goal	
Exceptions:		
Includes:		
Frequency of Use:	Once a day	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Retrieve Profile

Use Case ID:	133	
Use Case Name:	Retrieve Profile	
Created By:	Nathaniel Webber	Last Updated By:
Date Created:	09/24/20210	Last Revision Date:
Actors:	Adult with Physical Disability	
Description:	User Retrieves Profile	
Trigger:	User wants to create a Log in to their Profile	
Preconditions:	1. User must have a device with the program on it (PC, Tablet, etc.) 2. User must have already created a Profile	
Postconditions:	1. User has a Logged into their profile	
Normal Flow:	1. System Prompts User for Email 2. User Enters Email 3. System prompts user for Password 4. User enters Password 5. User Selects "Log in" 6. System Validates User 7. System Logs in User	
Alternative Flows:	[Alternative Flow 1 – Not in Network] 3a. In step 3 of the normal flow, if the User enters invalid Email or Password 1. System prompts User saying Email or Password was not correct 2. System prompts User to Re-enter Email and Password 3. System prompts user to reset password if they want to, or create a new profile	
Exceptions:		
Includes:		
Frequency of Use:	Once a day	
Special Requirements:	User must have a Profile	
Assumptions:	User understands English and has basic computer competency	
Notes and Issues:	1.	

Caregiver Use Cases

Deactivate Own Account

Use Case ID:	31		
Use Case Name:	Deactivate Own Account		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/20	Last Revision Date:	2020/12/11
Actors:	All Caregiver Users		
Description:	If for whatever reason, the user no longer wants to have an active account to take on new clients but may still want to have some records in case they are needed in the future. This will deactivate but not fully delete the user's account.		
Trigger:	The user no longer wanting an active account in the program.		
Preconditions:	[List any activities that must take place, or any conditions that must be true before the use case can be started. Number each pre-condition.] 1. The user has an active account. 2. The user is currently logged in. 3. The user is at their home screen.		
Postconditions:	1. The users account is now deactivated and can no longer accept new clients till reactivated.		
Normal Flow:	1. The user clicks on the option button on the screen 2. The system takes them to the options screen where the user can update their account information 3. The user scrolls down to the bottom of the screen and clicks on the deactivate account button 4. The system prompts the user with a warning about deactivating their account 5. The user clicks the okay button 6. The system asks for the user's username and password 7. The user types in their username and password, then clicks ok 8. The system deactivates their account, telling the user their account has been deactivated 9. The user clicks ok 10. The system sends the user to the login screen		
Alternative Flows:	5a. In step 5 of the normal flow, if the user clicks the no button		

[Alternative Flow 1 – Not in Network]	<p>1. The user clicks no on deactivating their account.</p> <p>2. Return to normal flow step 2.</p> <p>7a. In step 7 of the normal flow, if the user clicks the cancel button on the password and username popup</p> <ol style="list-style-type: none"> 1. The user clicks the cancel button instead of typing out their username and password 2. Use Case resumes on step 2 of normal flow
Exceptions:	<p>8a. In step 8 of the normal flow, If the user misspells their username and or password</p> <ol style="list-style-type: none"> 1. The system tells the user that they have misspelled their username and or password while clearing the password input box 2. User reenters their password and fixes their username if necessary. Then click the ok button 3. Use Case resumes on step 8 of normal flow
Includes:	Parts of this use case may be found in updating profile
Frequency of Use:	At least once or not at all during the life of an active account. Could be multiple times if a user is deactivating and reactivating their account multiple times.
Special Requirements:	No known unique requirements
Assumptions:	No known assumptions
Notes and Issues:	No known notes or issues

Delete Account

Use Case ID:	32		
Use Case Name:	Delete Account		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/20	Last Revision Date:	2020/12/11
Actors:	All Caregiver users		
Description:	The user no longer wants to have an account and no longer wants to keep any records from it. This should permanently delete their account from the system		
Trigger:	The user no longer wishes to have their account or would like to remake their account from scratch		
Preconditions:	<ol style="list-style-type: none"> 1. The users account is already deactivated 2. The user is at the login screen 		
Postconditions:	<ol style="list-style-type: none"> 1. The users account is permanently deleted. 		
Normal Flow:	<ol style="list-style-type: none"> 1. The system is displaying the login screen 2. The user types in their username and password 3. The user clicks the login button 4. The system displays a notification saying that the users account is deactivated and would like to know if they would like to reactivate it or delete it. 5. The user clicks the delete account button 6. The system asks for the username, password, and email of the account that is to be deleted. 7. The user types in the username, password, and email of their account. 8. The user clicks the ok button 9. The system will tell the user that the account was deleted 		
Alternative Flows:	4a. In step 4 of the normal flow, if the user clicks the reactivate button		
[Alternative Flow 1 – Not in Network]	<p>1. The system will tell the user that their account has been reactivated.</p> <p>2. To reattempt removing the account the user must go through the process of deactivating it again and then go to normal flow step 1.</p> <p>4b. in step 4 of the normal flow, if the user clicks the cancel button</p> <p>1. The user clicks the cancel button</p>		

	<p>2. Return to normal flow step 1</p> <p>8a. In step 8 of the normal flow, if the user clicks cancel instead of ok</p> <ol style="list-style-type: none"> 1. Return to normal flow step 1
Exceptions:	<p>3a. In step 3 of the normal flow, if the user misspells their username and or password</p> <ol style="list-style-type: none"> 1. The system will tell the user they have entered the incorrect username and or password, while removing the inputted password and leaving the username in the input box 2. The user retypes in the password and fixes the username if necessary 3. Return to normal flow step 3 <p>8a. In step 8 of the normal flow, if the user misspells the username, password and or email</p> <ol style="list-style-type: none"> 1. The system will tell the user they have entered the incorrect username, password, and or email while removing the inputted password 2. The user will retype in the password and fix up the username and or email if necessary 3. Return to normal flow step 8.
Includes:	<p>All the deactivate account use case is needed for this use case</p> <p>This use case also shows how to reactivate a deactivated account</p>
Frequency of Use:	At least once per account. Should not be too frequent
Special Requirements:	No known unique requirements
Assumptions:	<ol style="list-style-type: none"> 1. if the user has deactivated their account then it will not let them access their account
Notes and Issues:	<ol style="list-style-type: none"> 1. No known notes or issues

Update Account

Use Case ID:	33		
Use Case Name:	Update Account		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: User, Admin, Caretaker Secondary: System, database		
Description:	The user updates information on their profile.		
Trigger:	The user selects the Update Profile button in the user profile section.		
Preconditions:	1. The user is logged in 2. The user has found the user profile section 3.		
Postconditions:	1. The user's account has been updated 2. The system updates the display to show the user's profile information with the updated information.		
Normal Flow:	1. The user selects update account. 2. The system updates the display to allow each line of the profile to be edited. 3. The user updates each item as needed. 4. The user selects update account after all edits have been made. 5. The system logs the previous account information. 6. The system updates the display to show the account information with the updates saved on it without the ability to edit them. 7. The system alerts the user that the update has been made. 8. The user acknowledges the alert. 9. End of Normal Flow		
Alternative Flows:	4. In Normal Flow step 4, If the user decides not to update their account:		
[Alternative Flow 1 – Not in Network]	1. The user selects the cancel button instead of the update account button. 2. The system alerts the user that there were no updates made. 3. The user acknowledges the update.		

	4. The system updates the display to show the account information without being able to edit as it was before this function was invoked.
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	This function is intended to be used infrequently.
Special Requirements:	The database has a place to store the old and new account information.
Assumptions:	The user has the information on hand to update their account with.
Notes and Issues:	<ul style="list-style-type: none"> 1. Should the user have to update everything if they only want to update one item, or can they select updates individually? 2. What items should not be allowed to be updated?

Deny Given Permissions

Use Case ID:	34	
Use Case Name:	Deny Given Permissions	
Created By:	Ryan Taylor	Last Updated By: Ryan Taylor
Date Created:	2020/09/20	Last Revision Date: 2020/12/11
Actors:	All Caregiver user	
Description:	If the user does want the task, they can accept it	
Trigger:	The admin has sent a caregiver a permission to do something	
Preconditions:	1. The admin of a client has sent the user a permission they want them to do 2. The user is looking at their home screen	
Postconditions:	1. The user does get the permission 2. The admin gets notified that the user does want that permission	
Normal Flow:	1. The user sees a notification on their home screen 2. The user clicks on the notification 3. The system tells the user what permission the admin wants you to have 4. The user clicks accept on the permission 5. The system sends the answer to the admin	
Alternative Flows:	4a. In step 4 of the normal flow, if the user clicks deny	
[Alternative Flow 1 – Not in Network]	1. The user clicks the deny button 2. Use Case resumes on step 5 while getting the permission	
	4b. In step 4 of the normal flow, if the user clicks cancel	
	1. The system takes the user back to the home screen 2. Use Case resumes on step 1 of normal flow	
Exceptions:	There are no known exceptions	
Includes:	Parts of the accept given permissions can be found in this use case	
Frequency of Use:	Whenever an admin gives a caregiver a permission, so frequent	
Special Requirements:	The user has a client they are helping	
Assumptions:	1. A person should be able to accept or deny a permission instead of just giving the user the permission asap	

Notes and Issues:	1. No known notes or issues
--------------------------	-----------------------------

Creating Account

Use Case ID:	35		
Use Case Name:	Creating Account		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/18	Last Revision Date:	2020/12/11
Actors:	All users Guest Account, Certified helpers / Advise and review, Represent user / Review Only, and Support users		
Description:	The user wants to create an account to help the client. By doing this the users will have a usable account.		
Trigger:	The user wishes to create an account		
Preconditions:	1. The user does not have an account wishes to create one 2. The user is at the login screen of the program		
Postconditions:	1. The user's information is now recorded in the system for their account 2. The user now has an active account.		
Normal Flow:	1. The user clicks the create account button. 2. The program prompts the user with a form to fill out. 3. The user fills out their username, password, confirm password, email, confirm email, and optional phone number. 4. The user clicks create your account. 5. The system will tell the user that their account was created. 6. The user will acknowledge that their account was created by clicking the ok button. 7. The system will take the user to the log in screen.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	3a. In step 4 of the normal flow, if the user decides to not make an account 1. The user clicks the cancel button 2. System will ask the user if they wish to cancel and that everything typed in will be removed <ul style="list-style-type: none"> a. The user clicks no. b. They system returns the user to the form they were filling out. c. Return to Normal flow 3. 3. The user clicks yes. 4. The system removes all the information the user has put into the form and brings the user to the login screen.		
Exceptions:	4a. In step 4 of the normal flow, if the password and confirm password do not match		

	<ol style="list-style-type: none"> 1. System will remove what the user put into the password and confirm password options 2. System will tell the user that the password and confirm password options do not match and needs to be redone Customer enters correct PIN 3. The user fills in the password and confirm password options 4. Use Case resumes on step 4 <p>4b. In step 4 of the normal flow, if the email and confirm email do not match</p> <ol style="list-style-type: none"> 1. System will remove what the user put into the email and confirm email options 2. System will tell the user that the email and confirm email options do not match and needs to be redone 3. The user fills in the email and confirm email options 4. Use Case resumes on step 4
Includes:	None
Frequency of Use:	Every time the user wishes to make a new account, in a perfect world only once but can be multiple times if the user repeatedly deletes and makes an account
Special Requirements:	None
Assumptions:	That when the program is started up it will bring the user to the login screen
Notes and Issues:	<ol style="list-style-type: none"> 1. What is the maximum size of the password? 2. What is the maximum size of the username? 3. Will all special characters be accepted within the username and or password?

Forgot Login Information

Use Case ID:	36		
Use Case Name:	Forgot Login Information		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: User/Admin/Caretakers Secondary:		
Description:	The user forgot their login information and needs help to retrieve and/or reset it		
Trigger:	The user has selected the Forgot username/password button.		
Preconditions:	1. The user has selected the forgot username/password button.		
Postconditions:	1. The user has been given the ability to reset their account password. 2. The system returns to the login page.		
Normal Flow:	1. The system prompts the user to enter their email. 2. The user enters their email into the expected field. 3. The user selects the enter button. 4. The system verifies their email. 5. The system prompts the user to answer their security questions. 6. The user answers the security questions. 7. The user selects the continue button. 8. The system verifies the security questions. 9. The system prompts the user to enter a new username and email. 10. The user enters the new username and password into the expected fields. 11. The user selects enter. 12. The system alerts the user that their username and password have been updated. 13. The user acknowledges the alert. 14. The system displays the login page. 15. End of Normal Flow.		
Alternative Flows:	3 In step 3 of the normal flow, if the system cannot verify their email: 1. The system alerts the user that their email could not be found. 2. The user acknowledges the alert. 3. Use Case resumes Normal Flow step 1.		

	<p>6. In step 6 of the normal flow, if the system could not verify the users answers to the security questions:</p> <ol style="list-style-type: none"> 1. The system alerts the user that some of their security question answers are incorrect. 2. The user acknowledges the alert. 3. Use case resumes Normal Flow step 5.
Exceptions:	6. In step 6, If the user cannot answer their security questions correctly
Includes:	[List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g., steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.]
Frequency of Use:	This function should have minimal use so long as users can remember their login information.
Special Requirements:	The system must be connected to the database to confirm the answers to the security questions and to reset the username and password.
Assumptions:	<p>The create account use case includes security questions.</p> <p>The username is not the user’s email.</p>
Notes and Issues:	<p>1. What to do if the user cannot answer their security questions correctly?</p>

Leave Note on Wall

Use Case ID:	37		
Use Case Name:	Leave Note on Wall		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: Admin, Caretakers Secondary: system, database		
Description:	The user leaves a note on the User's Wall		
Trigger:	The user selects leave note option on a User's Wall.		
Preconditions:	1. The user is logged in 2. The user is connected to a User's account 3. The user has permissions to view the User's wall.		
Postconditions:	1. The user's note is posted beneath the User's wall. 2. The system alerts the user that their note has been left.		
Normal Flow:	1. The user clicks the leave note button on a User's wall. 2. The system prompts the user to write a message in the text box. 3. The user types their message. 4. The user selects the leave note button. 5. The user acknowledges the alert. 6. The user's note is posted on the user's wall. 7. The system updates the display to show their note under the User's wall. 8. End of Normal Flow		
Alternative Flows:	3. In step 3 or the Normal Flow if the user decides not to leave their note: 1. The user selects the cancel button instead of the leave note button. 2. The system deletes their text and removes the text box. 3. The display is updated to show the User's wall. 4. The system alerts the user that their note was not left 5. End of Alternate Flow		
Exceptions:	N/A		
Includes:	N/A		

Frequency of Use:	May never be used by certain users. Others may use it dozens of times or even more.
Special Requirements:	The database must have a place to store the notes and a way to sort them depending on what users should be able to see what notes.
Assumptions:	The user is leaving a note that is relevant to the User.
Notes and Issues:	<ol style="list-style-type: none"> 1. Should a user be able to leave a reply directly on another user's note, like a little thread beneath the first note? 2. Should there be a character limit to a note? 3. Should there be any restrictions on language left?

User Login

Use Case ID:	38		
Use Case Name:	User Login		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary Actor: User/Admin/Caretakers Secondary Actor: the system, database		
Description:	The user would log into their account to interact with our system. This would allow them to see their account information and to be able to complete tasks, such as the activities, marking off tasks, leaving and reading notes, and logging out.		
Trigger:	The user has a need to access the application.		
Preconditions:	1. The user has already created an account 2. The user has the application open.		
Postconditions:	1. The user has successfully logged into the application. 2. The user is now viewing the home page for their account.		
Normal Flow:	1. The system displays an option to login. 2. The user selects the login button. 3. The system prompts the user for their username and password with a login button below these fields. 4. The user enters their username and password into the expected fields. 5. The user selects Login. 6. The system verifies the user's credentials with the database. 7. Once the user is confirmed, the system displays a log in message to the user. 8. The system finds the user's profile within the database. 9. The system finds the correct home page for the user. With the database. 10. The system displays the user's home page once found. 11. End of Normal Flow.		
Alternative Flows:	6a. In step 6, If the system cannot verify the user in the database: 1. The system alerts the user that their username or password is not correct. 2. The user acknowledges the alert. 3. The system returns to Normal Flow step 3. 6b. In step 6, If the user's account is currently deactivated:		

	<p>1. The system alerts the user that their account is deactivated and cannot be logged into at this time.</p> <p>2. The user acknowledges the alert.</p> <p>3. Use case resumes step one. The user can invoke the reactivate account use case (UC_Reactivate_Account)</p>
Exceptions:	<p>4. In step 4, If the user does not remember their username or password:</p> <ol style="list-style-type: none"> 1. The user will select the “Forgot Username/ Password” button. (UC_Forgot_Login_Information_Z12) 2. The system invokes the forgot login information use case. 3. The current Login use case ends.
Includes:	<p>UC_Forgot_Login_Information_Z12</p> <p>UC_Reactivate_Account</p>
Frequency of Use:	Frequency of use will depend on the user. May be used only a handful of times in their entire use of the application while another user may use this a dozen times in a day.
Special Requirements:	The system must be able to find user accounts based on the user's username and password.
Assumptions:	<p>The user understands English.</p> <p>The application is connected to the database in some way.</p> <p>The user has an expectation to get into their account.</p>
Notes and Issues:	<ol style="list-style-type: none"> 1. What is the username made up from? Some user created username or email? 2. What would be restrictions on the password be? Six characters, or an unusual character, etc. 3. Does the user have to click “Login” to get the username and password prompts or will the first screen have the fields open right away? 4. Is there going to be an auto login or remember me choice?

User Logout

Use Case ID:	39				
Use Case Name:	User Logout				
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette		
Date Created:	2020/09/18	Last Revision Date:	2020/09/18		
Actors:		Primary Actor: User/Admin/Caretakers Secondary Actor: the system, database			
Description:	The user would like to log out of their account once they have completed their tasks within the application. This protects the user's profile from anyone else who would gain access to the user's device from getting into their account.				
Trigger:	The user no longer has immediate need for the application and would like to close it.				
Preconditions:	<ol style="list-style-type: none"> 1. The user is already logged into their account. 2. The user no longer wants to be logged into their account. 				
Postconditions:	<ol style="list-style-type: none"> 1. The user is successfully logged out of their account. 2. The user is looking at the log in/ opening screen. 				
Normal Flow:	<ol style="list-style-type: none"> 1. The system will have a logout button displayed in the help menu. 2. The user will select the logout button. 3. The system will check if there is any unsaved work. (Check_For_Unsaved_Work) 4. If everything is saved, the system will log the user out of their account. 5. The system will display the login screen. 6. The system will alert the user that they have been logged out of their account. 7. End of Normal Flow. 				
Alternative Flows:	<p>3. In step 3 of the normal flow, if there is any unsaved work:</p> <ol style="list-style-type: none"> 1. The system alerts the user that there is unsaved work and logging out may result in the loss of any unsaved work. 2. The system will prompt the user to select an option to either continue to log out or to cancel logging out. 3. The user will select an option. <ul style="list-style-type: none"> a. Cancel would cancel the logout function and End Alternate Flow b. Continue would abandon or save any unsaved work depending on the circumstances. 4. Use case resumes on Step 5. 				

	<p>Exceptions: 3. In Alternate Flow Step 3b. If the system could not abandon any unsaved work:</p> <ol style="list-style-type: none"> 1. The system would alert the user that they must complete the task that cannot be abandoned before logging out.
	<p>Includes: Check_For_Unsaved_Work</p>
	<p>Frequency of Use: The user would use this button either as many times as they use the Login function or just a few times depending on who has access to their device and if they mind others having access to their account.</p>
	<p>Special Requirements: The system must be able to close a user's account to complete the logout process.</p>
	<p>Assumptions: The user understands English. The application has not lost connection to the database. The user has some Internet literacy to know of the logout button and where it might be found.</p>
	<p>Notes and Issues:</p> <ol style="list-style-type: none"> 1. Would the system auto logout the user if the application is closed?

Reactivate Note

Use Case ID:	40		
Use Case Name:	Reactivate Note		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: Admin, Caretakers Secondary: system, database		
Description:	The user Reactivates a note they created		
Trigger:	The user selects edit note option next to their note		
Preconditions:	1. The user is logged in 2. The user can see their note in the Deactivated notes section of their profile.		
Postconditions:	1. The user's note is Reactivated 2. The system alerts the user that their note has been Reactivated		
Normal Flow:	1. The user clicks the edit note button on their note to be Reactivated. 2. The user selects the Reactivate note button. 3. The system alerts the user that their note will be Reactivated. 4. The user acknowledges the alert. 5. The user's note is Reactivated. 6. The Reactivated note moves to the Reactivated notes section of their profile. 7. The system updates the display without their Reactivated note. 8. End of Normal Flow		
Alternative Flows:	2. In step 2 or the Normal Flow if the user decides not to Reactivate their note: 1. The user selects the cancel button instead of the Reactivate note button. 2. The display is updated to show the page before this function was invoked. 3. The system alerts the user that their note was not Reactivated 4. End of Alternate Flow		
Exceptions:	N/A		
Includes:	N/A		

Frequency of Use:	May never be used by certain users. Intended to be a functionality that is rarely used.
Special Requirements:	The database must have a place to store the notes and a way to sort them depending on what users should be able to see what notes.
Assumptions:	The user has reason to Reactivate their note
Notes and Issues:	<ol style="list-style-type: none"> 1. Should the user have to get permission from the admin to check if they can reactivate a note? 2. Should the system double check to see if they still have permissions to leave a note on the User's account where it was originally posted at?

Request Permissions

Use Case ID:	41		
Use Case Name:	Request Permissions		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Caretakers		
Description:	If a caretaker believes they should have more access to the user's profile and/or to the administrative functions involved with the application, the caretaker can request they be given more access and functionality.		
Trigger:	The user selects the "Request permission" button.		
Preconditions:	1. The user is logged in 2. The user is not the Admin.		
Postconditions:	1. The Admin is sent a request for permissions. 2. The system displays a request sent message.		
Normal Flow:	1. The user selects the Request permissions button. 2. The system displays a form and prompts the user to enter what permissions they are requesting from the Admin. 3. The user enters the required information. 4. The user selects send. 5. The system sends the request to the admin. 6. The system returns to the home page. 7. The system alerts the user that their request has been sent. 8. The user acknowledges alert. 9. End of Normal Flow. 10.		
Alternative Flows:	4. In step 4 of the normal flow, if the user did not enter anything: 1. The system alerts the user that they need to fill out the form before the request can be sent. 2. The user acknowledges the alert. 3. Use case resumes step 2.		
Exceptions:	7. In step 7 of the normal flow, if the user attempts to request the same permissions as they already have: 1. The system alerts the user that they have already requested some or all the permissions requested.		

	2. Use case resumes Normal Flow step 2
Includes:	Send_Permission_Request_To_Admin
Frequency of Use:	This function may never be used at all but could be used up to the number of functions that the system has that could be given to another user.
Special Requirements:	<p>The user requesting permissions does not already have all the permissions an admin could give.</p> <p>The user is not blocked from using this feature.</p> <p>The user is not requesting the same permissions</p>
Assumptions:	User is aware that they may need more functionality.
Notes and Issues:	<ol style="list-style-type: none"> 1. Does the request form an empty form or a check box? <ol style="list-style-type: none"> a. If it is a checkbox, should the user be able to see all the functionality that is not included in their current view? 2. Should the user be allowed to spam this button? 3. Should every user be allowed to see this functionality? 4. Can the Admin disable this from any users view? 5. Can the Admin mute users where they see this function, but the Admin does not actually get the requests? 6. If the user has already sent a request, should they be allowed to send another until the first one is resolved?

Edit Note

Use Case ID:	42		
Use Case Name:	Edit Note		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
	Primary: Admin, Caretakers Actors: Secondary: system, database		
Description:	The user edits a note they created		
Trigger:	The user selects edit note option next to their note		
Preconditions:	1. The user is logged in 2. The user can see their note		
Postconditions:	1. The user's note is edited 2. The system alerts the user that their note has been edited		
Normal Flow:	1. The user clicks the edit note button on their note to be edited. 2. The system prefills a text box with their previous note. 3. The system prompts the user to edit their message in the text box. 4. The user edits their message. 5. The user selects the leave note button. 6. The user acknowledges the alert. 7. The user's old note is deactivated. 8. The user's note is edited. 9. The system updates the display to show their note. 10. End of Normal Flow		
Alternative Flows:	3. In step 3 or the Normal Flow if the user decides not to edit their note: 1. The user selects the cancel button instead of the leave note button. 2. The system removes the text box. 3. The display is updated to show the page before this function was invoked. 4. The system alerts the user that their note was not edited 5. End of Alternate Flow		
Exceptions:	N/A		
Includes:	N/A		

	Frequency of Use: May never be used by certain users. Others may use it dozens of times or even more. Intended to be a functionality that is rarely used.
Special Requirements:	The database must have a place to store the notes and a way to sort them depending on what users should be able to see what notes.
Assumptions:	The user has reason to edit their note
Notes and Issues:	<ol style="list-style-type: none"> 1. Should a user be able to edit a note left by a different user? 2. Should there be a character limit to a note? 3. Should there be any restrictions on language left? 4. Can the Admin edit anyone's note?

Leave a Note on a Task

Use Case ID:	43		
Use Case Name:	Leave a Note on a Task		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: Admin, Caretakers Secondary: system, database		
Description:	The user leaves a note on the User's task		
Trigger:	The user selects leave note option beneath a User's task.		
Preconditions:	1. The user is logged in 2. The user is connected to a User's account 3. The user has permissions to view the User's tasks		
Postconditions:	1. The user's note is posted beneath the User's task. 2. The system alerts the user that their note has been left.		
Normal Flow:	1. The user clicks the leave note button beneath a User's task 2. The system prompts the user to write a message in the text box. 3. The user types their message. 4. The user selects the leave note button. 5. The user acknowledges the alert. 6. The user's note is posted beneath the user's task. 7. The system updates the display to show their note under the User's task. 8. End of Normal Flow		
Alternative Flows:	3. In step 3 or the Normal Flow if the user decides not to leave their note: 1. The user selects the cancel button instead of the leave note button. 2. The system deletes their text and removes the text box. 3. The display is updated to show the User's task. 4. The system alerts the user that their note was not left 5. End of Alternate Flow		
Exceptions:	N/A		
Includes:	N/A		

Frequency of Use:	May never be used by certain users. Others may use it dozens of times or even more.
Special Requirements:	The database must have a place to store the notes and a way to sort them depending on what users should be able to see what notes.
Assumptions:	The user is leaving a note that is relevant to the User.
Notes and Issues:	<ol style="list-style-type: none"> 1. Should a user be able to leave a reply directly on another user's note, like a little thread beneath the task? 2. Should there be a character limit to a note? 3. Should there be any restrictions on language left?

Deactivate Note

Use Case ID:	44		
Use Case Name:	Deactivate Note		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: Admin, Caretakers Secondary: system, database		
Description:	The user deactivates a note they created		
Trigger:	The user selects edit note option next to their note		
Preconditions:	1. The user is logged in 2. The user can see their note		
Postconditions:	1. The user's note is deactivated 2. The system alerts the user that their note has been deactivated		
Normal Flow:	1. The user clicks the edit note button on their note to be deactivated. 2. The user selects the deactivate note button. 3. The system alerts the user that their note will be deactivated. 4. The user acknowledges the alert. 5. The user's note is deactivated. 6. The deactivated note moves to the deactivated notes section of their profile. 7. The system updates the display without their deactivated note. 8. End of Normal Flow		
Alternative Flows:	2. In step 2 or the Normal Flow if the user decides not to deactivate their note: 1. The user selects the cancel button instead of the deactivate note button. 2. The display is updated to show the page before this function was invoked. 3. The system alerts the user that their note was not deactivated 4. End of Alternate Flow		
Exceptions:	N/A		
Includes:	N/A		

Frequency of Use:	May never be used by certain users. Others may use it dozens of times or even more. Intended to be a functionality that is rarely used.
Special Requirements:	The database must have a place to store the notes and a way to sort them depending on what users should be able to see what notes.
Assumptions:	The user has reason to deactivate their note
Notes and Issues:	<ol style="list-style-type: none"> 1. Should a user be able to deactivate a note left by a different user? 2. Should there be a character limit to a note? 3. Should there be any restrictions on language left? 4. Can the Admin deactivate anyone's note?

Delete Note

Use Case ID:	45		
Use Case Name:	Delete Note		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: Admin, Caretakers Secondary: system, database		
Description:	The user Deletes a note they created		
Trigger:	The user selects edit note option next to their note		
Preconditions:	1. The user is logged in 2. The user can see their note in the deactivated notes section of their profile.		
Postconditions:	1. The user's note is Deleted 2. The system alerts the user that their note has been Deleted		
Normal Flow:	1. The user clicks the edit note button on their note to be Deleted. 2. The user selects the Delete note button. 3. The system alerts the user that their note will be Deleted. 4. The user acknowledges the alert. 5. The user's note is Deleted. 6. The system updates the display without their Deleted note. 7. End of Normal Flow		
Alternative Flows:	2. In step 2 or the Normal Flow if the user decides not to Delete their note: 1. The user selects the cancel button instead of the Delete note button. 2. The display is updated to show the page before this function was invoked. 3. The system alerts the user that their note was not Deleted 4. End of Alternate Flow		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	May never be used by certain users. Others may use it dozens of times or even more. Intended to be a functionality that is rarely used.		

Special Requirements:	The database must have a place to store the notes and a way to sort them depending on what users should be able to see what notes.
Assumptions:	The user has reason to Delete their note
Notes and Issues:	<ol style="list-style-type: none"> 1. Should a user be able to Delete a note left by a different user? 2. Should there be a character limit to a note? 3. Should there be any restrictions on language left? 4. Can the Admin Delete anyone's note?

Deny Invitation

Use Case ID:	46		
Use Case Name:	Deny Invitation		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: Caretakers Secondary: Admin		
Description:	The user is sent a request from the Admin of a User to join their group to see and/or interact with the User's profile.		
Trigger:	The Admin has sent the user a request to be associated with a User's profile.		
Preconditions:	1. The user has an account 2. The user is logged into their account 3. The Admin and User already have their accounts.		
Postconditions:	1. The user has denied the invitation to be linked to a User's account		
Normal Flow:	1. The system prompts the user that they have been invited to be linked to a User's account in the form of an overlay window. 2. The alert would include information about the User and Admin. 3. The alert would include their permission settings. 4. The alert the user to accept or deny the invitation. 5. The user selects deny. 6. The system sends a message to the Admin that the user has denied the invitation. 7. The system alerts the user that they have denied the invitation. 8. The system resumes the action that happened before this alert showed up. 9. End of Normal Flow		
Alternative Flows:	6. In step 6 of the Normal Flow, If the user selects accept: 1. Use case invokes UC_Accept_Invitation_Z12		
Exceptions:	1. Sd N/A		
Includes:	UC_Accept_Invitation_Z12		
Frequency of Use:	This will differ between users. A babysitter may see this infrequently, but a doctor's office or someone who would regularly interact with Users, like teachers might be sent many invitations. 1 – Dozens of uses.		

Special Requirements:	The user is not yet connected with the User in the invitation.
Assumptions:	The user has no need to be connected to this User's account.
Notes and Issues:	<ul style="list-style-type: none"> 1. Can the Admin force this connection, make it mandatory? 2. Should there be a remind me later function? 3. Should the user be able to request more permissions than those included in the invitation before accepting?

Accept Invitation

Use Case ID:	47		
Use Case Name:	Accept Invitation		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: Caretakers Secondary: Admin		
Description:	The user is sent a request from the Admin of a User to join their group to see and/or interact with the User's profile.		
Trigger:	The Admin has sent the user a request to be associated with a User's profile.		
Preconditions:	1. The user has an account 2. The user is logged into their account 3. The Admin and User already have their accounts.		
Postconditions:	1. The user has accepted the invitation to be linked to a User's account		
Normal Flow:	1. The system prompts the user that they have been invited to be linked to a User's account in the form of an overlay window. 2. The alert would include information about the User and Admin. 3. The alert would include their permission settings. 4. The alert the user to accept or deny the invitation. 5. The user selects Accept. 6. The system sends a message to the Admin that the user has Accepted the invitation. 7. The system alerts the user that they have Accepted the invitation. 8. The system resumes the action that happened before this alert showed up. 9. End of Normal Flow		
Alternative Flows:	6. In step 6 of the Normal Flow, If the user selects deny: 1. Use case invokes UC_Deny_Invitation_Z12		
Exceptions:	1. Sd N/A		
Includes:	UC_Deny_Invitation_Z12		
Frequency of Use:	This will differ between users. A babysitter may see this infrequently, but a doctor's office or someone who would regularly interact with Users, like teachers might be sent many invitations. 1 – Dozens of uses.		

Special Requirements:	The user is not yet connected with the User in the invitation.
Assumptions:	The user has some need to be connected to this User's account.
Notes and Issues:	<ul style="list-style-type: none"> 1. Can the Admin force this connection, make it mandatory? 2. Should there be a remind me later function? 3. Should the user be able to request more permissions than those included in the invitation before accepting?

Check Off Items From Daily Task List

Use Case ID:	48		
Use Case Name:	Check Off Items from Daily Task List		
Created By:	Jory A. Wernette	Last Updated By:	Jory A. Wernette
Date Created:	2020/09/18	Last Revision Date:	2020/09/18
Actors:	Primary: Caretaker, Admin Secondary: System, User		
Description:	[Provide a brief description of the reason for and outcome of this use case.]		
Trigger:	The user has completed a task from their task list and the caretaker needs to check the item as complete.		
Preconditions:	1. The caretaker is logged into their account. 2. The caretaker is connected to the User's profile. 3. The caretaker has permissions from the Admin to have access to the User's task list. 4. The caretaker has permissions from the Admin to mark tasks as complete. 5. The caretaker has the User's task list pulled up in front of them.		
Postconditions:	1. The task has been marked as complete		
Normal Flow:	1. The caretaker waits for the User to complete their scheduled task. 2. Upon confirming the task is complete, the caretaker would click the box next to the task. 3. The system would check to confirm the caretaker has the correct permissions to make this change. 4. The system would update the display to show a check mark inside the selected box. 5. The system would record a timestamp for this task and update the database. 6. The caretaker will repeat these steps as needed until task list is finished. 7. End of Normal Flow		
Alternative Flows:	4. In step 4, If the caretaker finds out the task is in fact not complete: 1. The caretaker would find the uncompleted task in the task list that is currently displayed with a check mark. 2. The caretaker would click the box next to the task.		

	<p>3. The system would update the display to show the box without the checkmark.</p> <p>4. The system would record a timestamp for this task and update the database.</p> <p>5. Use case resumes step 5 of the Normal Flow.</p>
Exceptions:	<p>1. If the caretaker has a User's task list open in front of them and the Admin revokes their permission to edit the task list:</p> <ul style="list-style-type: none"> a. The caretaker would click the box next to a completed task. b. The system would check to confirm the caretaker has the correct permissions to make this change. c. The system would alert the caretaker that they do not have permissions to make this change.
Includes:	N/A
Frequency of Use:	This is a use that could be invoked dozens of times in a day.
Special Requirements:	The system needs to be able to differentiate between someone who can view a User's task list and someone who can check off items from the list.
Assumptions:	The caretaker would only check off the complete box if they confirmed the User did the required task.
Notes and Issues:	<p>1. Should the system check permissions with every click in the off chance that an Admin would revoke permissions while they are using the system or just have it check when they try to access the feature?</p> <p>2. Should the system save with every click, like above, or should there be a Submit button at the bottom of the page that would save the form once?</p>

View All Notifications

Use Case ID:	49		
Use Case Name:	View all Notifications		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/19	Last Revision Date:	2020/12/11
Actors:	All Caregivers		
Description:	The user needs to look at all their notifications on their account. This use case should show the user past and present notification that they have.		
Trigger:	Pressing the notification hyperlink text.		
Preconditions:	1. The user has an account. 2. The user is logged in.		
Postconditions:	1. The user will see all their notifications both answered and unanswered from the past to present		
Normal Flow:	1. The system shows the user their home screen including all the clients they help. 2. The user clicks on the hyperlink text called notifications, which is above the list of most recent notifications. 3. The system displays a list of all notifications ordered by date received. The user can change the order by clicking on the different row title		
Alternative Flows: [Alternative Flow 1 – Not in Network]	2a. In step 2 of the normal flow, if the user accidentally clicks on a different button on screen 1. System will take the user to wherever that button was designed to take them 2. With the help of the system the user can back track to normal flow 2 or normal flow or 3		
Exceptions:	No known exceptions		
Includes:	Parts of this Use Case may be used in the use case view by topic		
Frequency of Use:	Can be used often to view all the user's notifications. A lot.		
Special Requirements:	No known unique requirements		
Assumptions:	1. The user will click on the notification hyperlinked text to get to all the notifications instead of accidentally clicking on another button or hyperlinked text.		
Notes and Issues:	1. No known notes or issues		

View Tasks by Topic

Use Case ID:	50		
Use Case Name:	View Tasks by Topic		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/20	Last Revision Date:	2020/12/11
Actors:	All Caregiver Users		
Description:	If there are a lot of tasks the user can view, they may need to find specific tasks by searching by a key word. This should bring back any viewable tasks that have the searched key word.		
Trigger:	Typing in a key word and clicking the search button.		
Preconditions:	1. The user has an account. 2. The user is entrusted with helping a client. 3. The user is logged in. 4. The user is currently looking at		
Postconditions:	1. The user now sees all viewable tasks that have that key word		
Normal Flow:	1. The user clicks on the search bar labeled search tasks and types in a key word 2. The user clicks on the search button 3. The system shows any viewable tasks that match the key word		
Alternative Flows: [Alternative Flow 1 – Not in Network]	2a. In step 2 of the normal flow, if the user misspelled the key word in the search bar 1. System sows the user tasks matching the misspelling but not the tasks the user is looking for 2. Use Case resumes on step 1 2b. In step 2 of the normal flow, if the user misspelled the key word in the search bar 1. System sows the user tasks matching the misspelling but not the tasks the user is looking for 2. The user clicks a button labeled clear search 3. The system returns the user to the full original list of tasks that they were looking at before using the search bar 4. Return to normal flow step 1		
Exceptions:	3a. In step 3 of the normal flow, if there are no tasks that match the key word		

	<ol style="list-style-type: none"> 1. The system tells the user that there are no tasks with that key word 2. The user clicks the okay button 3. The system returns the user to the original list of tasks 4. Return to normal flow step 1.
Includes:	Could be used to help with checking off daily tasks or view all tasks.
Frequency of Use:	A little less frequent as looking at the daily tasks and all tasks.
Special Requirements:	No unique requirements known
Assumptions:	<ol style="list-style-type: none"> 1. There is a clear button to send the user back to the original list of tasks they were looking at.
Notes and Issues:	No known issues or notes

View Client's Current Tasks

Use Case ID:	51		
Use Case Name:	View Clients Current Tasks		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/19	Last Revision Date:	2020/12/11
Actors:	All caregiver users		
Description:	The user needs to look over the client's tasks for today to make sure they are completing everything they need to do. The user will be able to see the client's tasks for today.		
Trigger:	Clicking on a specific client the user can help		
Preconditions:	1. The user has an account. 2. The user is entrusted with helping a client. 3. The user is logged in.		
Postconditions:	1. The user sees all the client's daily tasks.		
Normal Flow:	1. The system shows the user their home screen including all the clients they help. 2. The user clicks on one of the clients. 3. The system shows the user the clients display screen with the client's tasks for that day at the bottom of the screen.		
Alternative Flows: [Alternative Flow 1 – Not in Network]	2a. In step 2 of the normal flow, if the user accidentally clicks on a different button on screen 1. System will take the user to wherever that button was designed to take them 2. With the help of the system the user can back track to normal flow 3		
Exceptions:	No known exceptions		
Includes:	Parts of this use case can be used in view all, view by topic, and check off items from daily tasks use cases		
Frequency of Use:	This will happen every time the user goes to look at a client. Possibly at least one a day or week.		
Special Requirements:	No known unique requirements		
Assumptions:	1. When pulling up a client to look at the users first sees the tasks the client needs to do today. 2. When looking at all the clients tasks the tasks will be sorted in chronological order by when they should be completed		

	3. The client or clients the user is helping on constantly shown on the side of the screen like side tabs
Notes and Issues:	No known notes or issues

Deny Given Permissions

Use Case ID:	52	
Use Case Name:	Deny Given Permissions	
Created By:	Ryan Taylor	Last Updated By: Ryan Taylor
Date Created:	2020/09/20	Last Revision Date: 2020/12/11
Actors:	All Caregiver user	
Description:	If the user does not want to handle a task, they can deny it	
Trigger:	The admin has sent a caregiver a permission to do something	
Preconditions:	1. The admin of a client has sent the user a permission they want them to do 2. The user is looking at their home screen	
Postconditions:	1. The user does not get the permission 2. The admin gets notified that the user does not want that permission	
Normal Flow:	1. The user sees a notification on their home screen 2. The user clicks on the notification 3. The system tells the user what permission the admin wants you to have 4. The user clicks deny on the permission 5. The system sends the answer to the admin	
Alternative Flows:	4a. In step 4 of the normal flow, if the user clicks accept	
[Alternative Flow 1 – Not in Network]	1. The user clicks the accept button 2. Use Case resumes on step 5 while getting the permission	
	4b. In step 4 of the normal flow, if the user clicks cancel	
	1. The system takes the user back to the home screen 2. Use Case resumes on step 1 of normal flow	
Exceptions:	There are no known exceptions	
Includes:	Parts of the accept given permissions can be found in this use case	
Frequency of Use:	Whenever an admin gives a caregiver a permission, so frequent	
Special Requirements:	The user has a client they are helping	
Assumptions:	1. It would be great for a person to be able to accept or deny a permission instead of just giving the user the permission asap	

Notes and Issues:

1. No known notes or issues

Reactivate Account

Use Case ID:	53		
Use Case Name:	Reactivate Account		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/20	Last Revision Date:	2020/12/11
Actors:	All Caregiver users		
Description:	The user wants to reactivate their account to use it again.		
Trigger:	The user wants to reactivate their deactivated account		
Preconditions:	1. The user's account is already deactivated 2. The user is at the login screen		
Postconditions:	1. The user's account is reactivated and usable.		
Normal Flow:	1. The system is displaying the login screen 2. The user types in their username and password 3. The user clicks the login button 4. The system displays a notification saying that the user's account is deactivated and would like to know if they would like to reactivate it or delete it. 5. The user clicks the reactivate button account button 6. The system asks for the username, password, and email of the account that is to be reactivated. 7. The user types in the username, password, and email of their account. 8. The user clicks the ok button 9. The system takes the user to the home screen of their account		
Alternative Flows:	4a. In step 4 of the normal flow, if the user clicks the delete button		
[Alternative Flow 1 – Not in Network]	1. The system will tell the user that their account has been reactivated. 10. The system asks for the username, password, and email of the account that is to be removed. 2. The user clicks the cancel button 3. Return to normal flow step 1		
	4b. in step 4 of the normal flow, if the user clicks the cancel button		
	1. The user clicks the cancel button 2. Return to normal flow step 1		

	<p>8a. In step 8 of the normal flow, if the user clicks cancel instead of ok</p> <ol style="list-style-type: none"> 1. Return to normal flow step 1
Exceptions:	<p>3a. In step 3 of the normal flow, if the user misspells their username and or password</p> <ol style="list-style-type: none"> 1. The system will tell the user they have entered the incorrect username and or password, while removing the inputted password and leaving the username in the input box 2. The user retypes in the password and fixes the username if necessary 3. Return to normal flow step 3 <p>8a. In step 8 of the normal flow, if the user misspells the username, password and or email</p> <ol style="list-style-type: none"> 1. The system will tell the user they have entered the incorrect username, password, and or email while removing the inputted password 2. The user will retype in the password and fix up the username and or email if necessary 3. Return to normal flow step 8.
Includes:	All the deactivate account use case is needed for this use case
Frequency of Use:	At least once per account. Can be multiple times depending on how often the user deactivates their account
Special Requirements:	No known unique requirements
Assumptions:	<ol style="list-style-type: none"> 1. If the user has deactivated their account, then it will not let them access their account
Notes and Issues:	<ol style="list-style-type: none"> 1. No known notes or issues

Refresh Screen

Use Case ID:	54		
Use Case Name:	Refresh Screen		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/20	Last Revision Date:	2020/12/11
Actors:	All Caregiver users		
Description:	If the user wants to refresh the page to see if something new will appear		
Trigger:	The user wants to know if something new has been added to the page they are looking at		
Preconditions:	<ol style="list-style-type: none"> 1. The user has an account 2. The user is logged in 3. The user is looking at any screen they wish to refresh (not the login screen) 		
Postconditions:	<ol style="list-style-type: none"> 1. The user is now looking at the most up to date version of the screen they have refreshed 		
Normal Flow:	<ol style="list-style-type: none"> 1. The user is looking at a screen they wish to have refreshed 2. The user finds the refresh button on the screen and clicks it 3. The system displays the most up to date version of the screen the user was looking at, along with a popup saying that the screen was refreshed 		
Alternative Flows:	No known alternative flows		
[Alternative Flow 1 – Not in Network]			
Exceptions:	No known exceptions		
Includes:	No known included use cases		
Frequency of Use:	Used as often as the user wants to use it. Very frequent		
Special Requirements:	No known unique requirements		
Assumptions:	That the user wants a refresh button that can refresh every viewable screen		
Notes and Issues:	<ol style="list-style-type: none"> 1. You cannot refresh the login screen 		

Remove Client

Use Case ID:	55		
Use Case Name:	Remove Client		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/20	Last Revision Date:	2020/12/11
Actors:	All Caregiver users		
Description:	When a caregiver is no longer caring for a client and wishes to remove their information from their personal account		
Trigger:	A caregiver is no longer taking care of a client		
Preconditions:	1. The user is logged in 2. The user has records of a client they are no longer taking care of 3. The user is looking at the home screen		
Postconditions:	1. The user no longer has access to the client they are no longer caring for		
Normal Flow:	1. The user is viewing the home screen and clicks on the desired client they wish to remove 2. The system displays the selected client's daily tasks 3. The user scrolls to the bottom of the screen to find the remove client button and clicks the button 4. The system displays a warning about removing the client from the caregivers account while requesting the user's username and password 5. The user types in their username and password 6. The user clicks the ok button 7. The system takes the user back to their home screen with a popup saying that the client has been removed from the user's account 8. The system sends a message to the admin saying that the caregiver has removed the client from their account		
Alternative Flows:	6a. In step 6 of the normal flow, if the user clicks the cancel button		
[Alternative Flow 1 – Not in Network]	1. Use Case resumes on step 2		
Exceptions:	6a. In step 6 of the normal flow, the user misspells their username and or password. 1. The system tells the user that they have entered the incorrect username and or password, while removing the typed in password		

	<p>2. The user retypes in the password and fixes the username if necessary.</p> <p>3. Use Case resumes on step 6 of normal flow</p>
Includes:	Some of the normal flow is used in the viewing use cases
Frequency of Use:	At the very least once per client. Not very frequent.
Special Requirements:	No known unique requirements
Assumptions:	<p>1. People will eventually wish to remove clients from their personal accounts.</p> <p>2. The remove client button is found at the bottom of a "clients viewable" screen on the caregiver's account.</p>
Notes and Issues:	<p>1. No known notes or issues</p>

View All Activities

Use Case ID:	56		
Use Case Name:	View All Activities		
Created By:	Ryan Taylor	Last Updated By:	Ryan Taylor
Date Created:	2020/09/19	Last Revision Date:	2020/12/11
Actors:	Certified helpers, Representatives, Primary helpers, and Parole Officers		
Description:	The user needs to look at past and present activities the client has done to help rehabilitate themselves. This use case should show the user all past and present activities the client has recorded within the program that are completed or uncompleted.		
Trigger:	Pressing the view all button on the page the user must view what the client needs to do today.		
Preconditions:	<ol style="list-style-type: none"> 1. The user has an account. 2. The user is entrusted with helping a client. 3. The user can view all the tasks a client has done. 4. The user is logged in. 		
Postconditions:	<ol style="list-style-type: none"> 1. The user receives all the client's tasks and can order them either alphabetically or chronologically 		
Normal Flow:	<ol style="list-style-type: none"> 1. The system shows the user their home screen including all the clients they help. 2. The user clicks on one of the clients. 3. The system shows the user the selected clients' wall. 4. the user clicks on the task button at the bottom of the screen. 5. the system shows the user the selected client's daily tasks 6. The user scrolls down to the last task to find and click on the show past tasks button. 7. The system shows the user all the past and present tasks the client has in chronological order. 8. The user clicks the name title of the column. 9. The system shows the user the tasks in alphabetical order. 10. The user clicks the date title of the column. 11. The system shows the user the tasks in chronological order. 12. the user clicks the done title of the column 13. the system shows the user the tasks in order of not completed to completed 		
Alternative Flows:	2,4a. In step 2 and 4 of the normal flow, if the user accidentally clicks on a different button on screen		

<p>[Alternative Flow 1 – Not in Network]</p>	<p>1. System will take the user to wherever that button was designed to take them 2. With the help of the system the user can backtrack to normal flow 3 or normal flow or 5</p> <p>6,8a. In step 6 and 8 of the normal flow, if the user accidentally clicks the cancel button</p> <ol style="list-style-type: none"> 1. The user clicks on the cancel button 2. Return to normal flow step 3 <p>6,8b. in step 6 and 8 of the normal flow, if the user accidentally clicks on a client</p> <ol style="list-style-type: none"> 1. The user clicks on a client's tab 2. Return to normal flow step 3
<p>Exceptions:</p>	<p>No known exceptions</p>
<p>Includes:</p>	<p>Parts of this Use Case may be used in the use case view by topic</p>
<p>Frequency of Use:</p>	<p>Can be used often among the users who have access to view all the client's tasks. At least one a day or week.</p>
<p>Special Requirements:</p>	<p>No known unique requirements</p>
<p>Assumptions:</p>	<ol style="list-style-type: none"> 1. When pulling up a client to look at the users first sees the tasks the client needs to do today. 2. When looking at all the clients tasks the tasks will be sorted in chronological order by when they should be completed 3. There is a button to sort by chronological order, and alphabetical order 4. The client or clients the user is helping on constantly shown on the side of the screen like side tabs
<p>Notes and Issues:</p>	<p>What happens when a user is helping many clients, like more than can fit on his home screen?</p>

Administrative Use Cases

Create Account

Use Case ID:	103		
Use Case Name:	Create Account		
Created By:	William Clark	Last Updated By:	
Date Created:	09/24/2020	Last Revision Date:	
Actors:	Administrator Data Store		
Description:	A user wishes to create an account to access user features as a top-level administrator.		
Trigger:	The user opens the application and selects [Create Account] from the initial interface.		
Preconditions:	1. Administrator has a valid email address.		
Postconditions:	1. Administrator successfully creates an account with which associated data will be stored by the data store. 2. Administrator is returned to the initial interface.		
Normal Flow:	1. Administrator opens the application and selects [Create Account] from the initial interface. 2. System prompts administrator to enter necessary data for the creation of an account (email, username, password). 3. System validates that an account is not already associated with the supplied information. 4. System prompts user to select an account type. 5. Administrator selects Administrator account type. 6. System creates a record of the account with the supplied information. 7. System notifies Administrator of success. 8. System prompts Administrator to log in with their created account, or to exit to main interface.		
Alternative Flows: Alternative Flow 1 – Information is already associated with Account	Document legitimate branches from the main flow to handle special conditions (also known as extensions). For each alternative flow reference the branching step number of the normal flow and the condition which must be true for this extension to be executed. e.g., Alternative flows in the <i>Withdraw Cash</i> transaction:		

	<p>3a. In step 3 of the normal flow if the information is already associated with an account in the data store.</p> <ol style="list-style-type: none"> 1. System will prompt Administrator to either enter different information or recover the account that was found. 2. Administrator opts to enter different account information. 3. Use Case resumes on step 2 <p>3b. In step 3 of the normal flow if the information is already associated with an account in the data store.</p> <ol style="list-style-type: none"> 1. System will prompt Administrator to either enter different information or recover the account that was found. 2. Administrator opts to recover their account. 3. System prompts for some relevant verification. 4. Administrator enters this verification data. 5. System prompts to reset password. 6. Administrator enters new valid password. 7. System updates account record with new password hash. 8. System notifies Administrator of recovery success. 9. System returns Administrator to initial interface. <p>Note: Insert a new row for each distinctive alternative flow.]</p>
Exceptions:	2a. In step 2 of the normal flow, if the administrator enters invalid information <ol style="list-style-type: none"> 1. Account creation is not allowed 2. System 3. Customer enters correct PIN 4. Use Case resumes on step 3 of normal flow]
Includes:	
Frequency of Use:	Once per administrator account
Special Requirements:	
Assumptions:	The Administrator understands the differences between account types
Notes and Issues:	How will account recovery be handled?

Add Client

Use Case ID:	104		
Use Case Name:	Add Client		
Created By:	William Clark	Last Updated By:	
Date Created:	09/24/2020	Last Revision Date:	
Actors:	Administrator Data Store		
Description:	An Administrator wishes to view clients associated with their account to access user features as a top-level administrator.		
Trigger:	The user logs into an existing Administrator account.		
Preconditions:	<ol style="list-style-type: none"> 1. Administrator has a saved account. 2. Administrator has logged into their account. 3. The target client has a valid account. 		
Postconditions:	<ol style="list-style-type: none"> 1. Administrator associated data will be loaded and displayed on the [Manage Clients] screen. 2. Administrator is presented with the [Manage Clients] screen. 		
Normal Flow:	<ol style="list-style-type: none"> 1. Administrator selects [Manage Clients] 2. System displays a list of client/client groups associated with the administrator account. 3. Administrator selects [Add Client] 4. System prompts the Administrator to enter a valid email for a client account. 5. Administrator enters an email or account name for a client. 6. System sends a request to the client account 7. System notifies Administrator that the request has been sent to the client account. 8. Administrator is presented with the [Manage Clients] screen. 		
Alternative Flows: The user wishes to add a client to a client group	<p>3. In step 3 of normal flow, if the user is going to add the client to an existing client group.</p> <ol style="list-style-type: none"> 1. Administrator selects an existing client/client group. 2. System displays [Client View] screen. 3. Administrator selects [Add Client] to this group. 4. Use Case resumes on step 4. 		
Exceptions:			
Includes:			

Frequency of Use:	1 per week at maximum
Special Requirements:	
Assumptions:	
Notes and Issues:	

Remove Client

Use Case ID:	105	
Use Case Name:	Remove Client	
Created By:	William Clark	Last Updated By:
Date Created:	09/24/2020	Last Revision Date:
Actors:	Administrator Data Store	
Description:	An Administrator wishes to remove a client associated with their account.	
Trigger:	The user logs into an existing Administrator account.	
Preconditions:	1. Administrator has a saved account. 2. Administrator has logged into their account. 3. The target client has a valid account.	
Postconditions:	1. Administrator associated data will be loaded and displayed on the [Manage Clients] screen. 2. Administrator is presented with the [Manage Clients] screen.	
Normal Flow:	1. Administrator selects [Manage Clients] 2. System displays a list of client/client groups associated with the administrator account. 3. Administrator selects an existing client/client group 4. System displays the [Client View]. 5. Administrator selects a single client. 6. Administrator selects [Remove Client] 7. System prompts for confirmation. 8. Administrator acknowledges. 9. System notifies Administrator that the request has been fulfilled. 10. Administrator is presented with the [Manage Clients] screen.	
Alternative Flows:		
Exceptions:		
Includes:		
Frequency of Use:	1 per week at maximum	
Special Requirements:		
Assumptions:		
Notes and Issues:		

Add Caregiver

Use Case ID:	106	
Use Case Name:	Add Caregiver	
Created By:	William Clark	Last Updated By:
Date Created:	09/24/2020	Last Revision Date:
Actors:	Administrator Data Store	
Description:	An Administrator wishes to add a caregiver account to a client account.	
Trigger:	The user logs into an existing Administrator account.	
Preconditions:	<ul style="list-style-type: none"> 1. Administrator has a saved account. 2. Administrator has logged into their account. 3. The target client has a valid account. 	
Postconditions:	<ul style="list-style-type: none"> 1. Administrator associated data will be loaded and displayed on the [Manage Clients] screen. 2. Administrator is presented with the [Manage Clients] screen. 	
Normal Flow:	<ul style="list-style-type: none"> 1. Administrator selects [Manage Clients] 2. System displays a list of client/client groups associated with the administrator account. 3. Administrator selects an existing client/client group 4. System displays the [Client View]. 5. Administrator selects a single client. 6. Administrator selects [Add Caregiver]. 7. System prompts the Administrator to enter a valid email for a client account. 8. Administrator enters an email or account name for a client. 9. System sends a request to the client account 10. System notifies Administrator that the request has been sent to the client account. 11. Administrator is presented with the [Manage Clients] screen. 	
Alternative Flows:		
Exceptions:		
Includes:		
Frequency of Use:	1 per week at maximum	
Special Requirements:		

Assumptions:
Notes and Issues:

Remove Caregiver

Use Case ID:	107	
Use Case Name:	Remove Caregiver	
Created By:	William Clark	Last Updated By:
Date Created:	09/24/2020	Last Revision Date:
Actors:	Administrator Data Store	
Description:	An Administrator wishes to remove a caregiver account to a client account.	
Trigger:	The user logs into an existing Administrator account.	
Preconditions:	<ol style="list-style-type: none"> 1. Administrator has a saved account. 2. Administrator has logged into their account. 3. The target client has at least one caregiver associated. 	
Postconditions:	<ol style="list-style-type: none"> 1. Administrator associated data will be loaded and displayed on the [Manage Clients] screen. 2. Administrator is presented with the [Manage Clients] screen. 	
Normal Flow:	<ol style="list-style-type: none"> 1. Administrator selects [Manage Clients] 2. System displays a list of client/client groups associated with the administrator account. 3. Administrator selects an existing client/client group 4. System displays the [Client View]. 5. Administrator selects a single caregiver associated with that client. 6. System displays [Caregiver Options]. 7. Administrator selects [Remove Caregiver]. 8. System prompts for confirmation. 9. Administrator acknowledges. 10. System notifies Administrator that the caregiver has been removed as a relation to the client. 11. Administrator is presented with the [Manage Clients] screen. 	
Alternative Flows:		
Exceptions:		
Includes:		
Frequency of Use:	1 per week at maximum	
Special Requirements:		
Assumptions:		

Notes and Issues:

Edit Caregiver Permissions

Use Case ID:	108	
Use Case Name:	Edit Caregiver Permissions	
Created By:	William Clark	Last Updated By:
Date Created:	09/24/2020	Last Revision Date:
Actors:	Administrator Data Store	
Description:	An Administrator wishes to edit a caregivers' permissions in relation to a client account.	
Trigger:	The user logs into an existing Administrator account.	
Preconditions:	<ul style="list-style-type: none"> 1. Administrator has a saved account. 2. Administrator has logged into their account. 3. The target client has at least one caregiver associated. 	
Postconditions:	<ul style="list-style-type: none"> 1. Administrator associated data will be loaded and displayed on the [Manage Clients] screen. 2. Administrator is presented with the [Manage Clients] screen. 	
Normal Flow:	<ol style="list-style-type: none"> 1. Administrator selects [Manage Clients] 2. System displays a list of client/client groups associated with the administrator account. 3. Administrator selects an existing client/client group 4. System displays the [Client View]. 5. Administrator selects a single caregiver associated with that client. 6. System displays [Caregiver Options]. 7. Administrator selects [Edit Permissions]. 8. System displays [Edit Permissions] screen. 9. Administrator changes permissions per the options menu provided. 10. Administrator selects [Save Changes] 11. System prompts for confirmation. 12. Administrator acknowledges. 13. System notifies Administrator that the caregiver has been edited in relation to the client. 14. Administrator is presented with the [Manage Clients] screen. 	
Alternative Flows: The administrator wishes to apply a previously	<p>9. In step 9 of normal flow if the user is going to apply a previously created permissions profile.</p> <p>1. Administrator selects an existing [Permissions Profile] from the list.</p> <p>2. Use Case resumes on step 10.</p>	

created permissions profile	
Exceptions:	
Includes:	
Frequency of Use:	1 per week at maximum
Special Requirements:	<p>The [Permissions Profile] includes preset options, including but not limited to:</p> <ol style="list-style-type: none"> 1. Caregiver Manager (can add or remove caregivers). 2. History Reviewer (can review the clients' performance history). 3. Goal Setter (can implement new goals for client). 4. Goal Adjuster (can edit individual goals but not add or remove them).
Assumptions:	
Notes and Issues:	

Create Permissions Profile

Use Case ID:	109	
Use Case Name:	Create Permissions Profile	
Created By:	William Clark	Last Updated By:
Date Created:	09/24/2020	Last Revision Date:
Actors:	Administrator Data Store	
Description:	An Administrator wishes to edit a caregivers' permissions in relation to a client account.	
Trigger:	The user logs into an existing Administrator account.	
Preconditions:	<ul style="list-style-type: none"> 1. Administrator has a saved account. 2. Administrator has logged into their account. 3. The target client has at least one caregiver associated. 	
Postconditions:	<ul style="list-style-type: none"> 1. Administrator associated data will be loaded and displayed on the [Manage Clients] screen. 2. Administrator is presented with the [Manage Clients] screen. 	
Normal Flow:	<ul style="list-style-type: none"> 1. Administrator selects [Manage Clients] 2. System displays a list of client/client groups associated with the administrator account. 3. Administrator selects [Settings] 4. System displays the [Settings] screen. 5. Administrator selects [Create Permissions Profile] 6. Administrator changes permissions per the options menu provided. 7. Administrator selects [Save Changes] 8. System prompts for confirmation. 9. Administrator acknowledges. 10. System notifies Administrator that the permissions profile has been saved for future use. 11. Administrator is presented with the [Manage Clients] screen. 	
Alternative Flows:		
Exceptions:		
Includes:		
Frequency of Use:	1 per month at maximum	

Special Requirements:
Assumptions:
Notes and Issues:

Deactivate Account

Use Case ID:	110		
Use Case Name:	Deactivate Account		
Created By:	William Clark	Last Updated By:	
Date Created:	09/24/2020	Last Revision Date:	
Actors:	Administrator Data Store		
Description:	An Administrator wishes to edit a caregivers' permissions in relation to a client account.		
Trigger:	The user logs into an existing Administrator account.		
Preconditions:	1. Administrator has a saved account. 2. Administrator has logged into their account.		
Postconditions:	1. Administrator is presented with the initial interface.		
Normal Flow:	1. Administrator selects [Settings] 2. System displays the [Settings] screen. 3. Administrator selects [Deactivate Account] 4. System prompts for confirmation. 5. Administrator acknowledges. 6. System notifies Administrator that the account has been deactivated, but data has been preserved for future use. 7. Administrator is presented with the initial interface.		
Alternative Flows: The user wishes to have data removed with account deactivation	2. In step 2 of normal flow, if the user wishes to have data removed along with account deactivation. 1. Administrator selects [Also Delete Account Data] 2. System prompts that this is not reversible. 3. Administrator acknowledges. 4. Use Case resumes on step 6.		
Exceptions:			
Includes:			
Frequency of Use:	1 per account		
Special Requirements:			
Assumptions:			
Notes and Issues:			

Log In

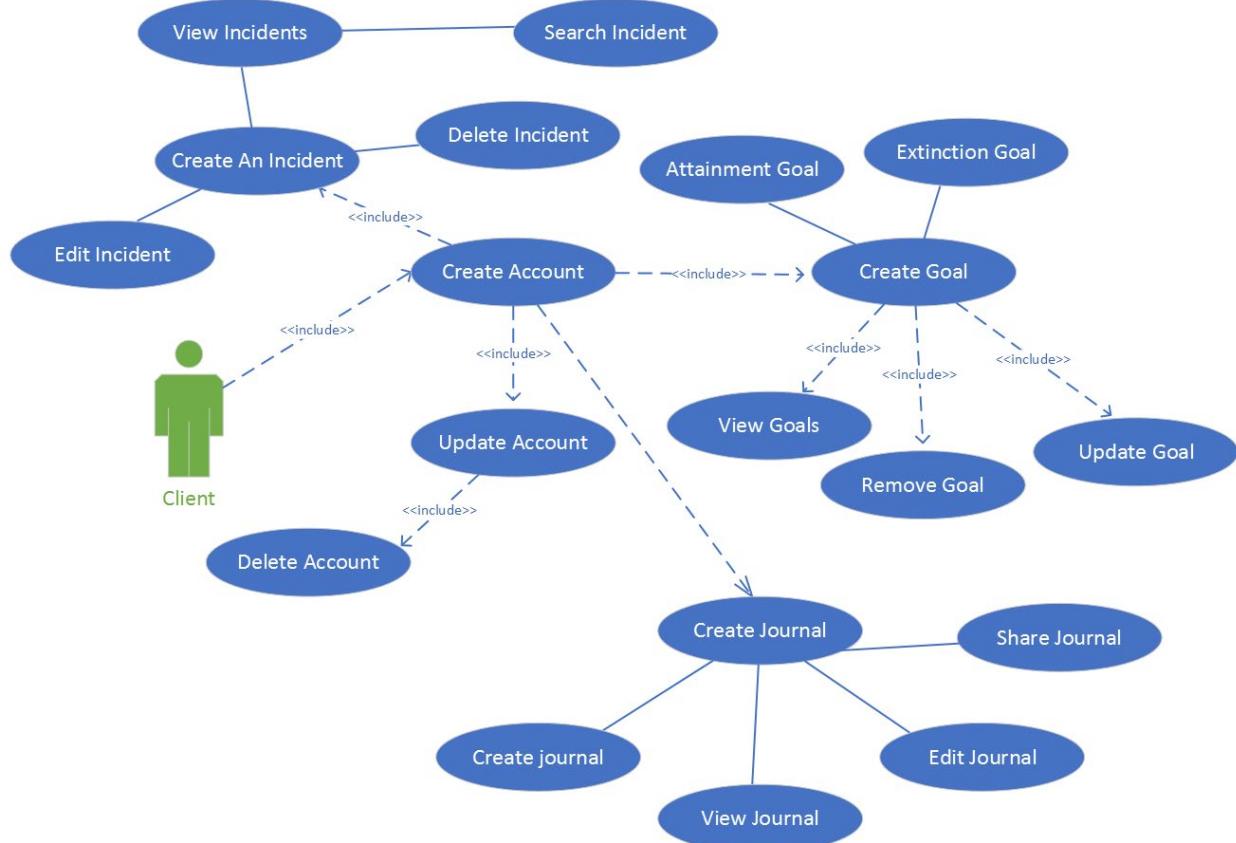
Use Case ID:	134	
Use Case Name:	Log In	
Created By:	William Clark	Last Updated By:
Date Created:	09/24/2020	Last Revision Date:
Actors:	Administrator Data Store	
Description:	An Administrator wishes to log in to an account to access user features as a top-level administrator.	
Trigger:	The user opens the application and selects [Log-In] from the initial interface.	
Preconditions:	1. Administrator has a saved account.	
Postconditions:	1. Administrator associated data will be loaded and displayed on the [Home] screen. 2. Administrator is presented with the [Home] screen.	
Normal Flow:	1. Administrator opens the application and selects [Log-In] from the initial interface. 2. System prompts administrator to enter necessary data authenticate an account (email/username, password). 3. Administrator enters credentials. 4. System validates that account credentials are valid. 5. System creates a record of the log in. 6. System notifies Administrator of success. 7. Administrator is presented with the [Home] screen.	
Alternative Flows:	4. In step 4 of the normal flow if the account credentials are not valid.	
Alternative Flow 1 – Account credentials are not valid	1. System will prompt Administrator to either enter different information or recover the account. 2. Administrator opts to enter different account information. 3. Use Case resumes on step 3.	
Exceptions:		
Includes:		
Frequency of Use:	One-seven times per day per user account.	
Special Requirements:		
Assumptions:		

Notes and Issues: How will account recovery be handled?

Use Case Drawings

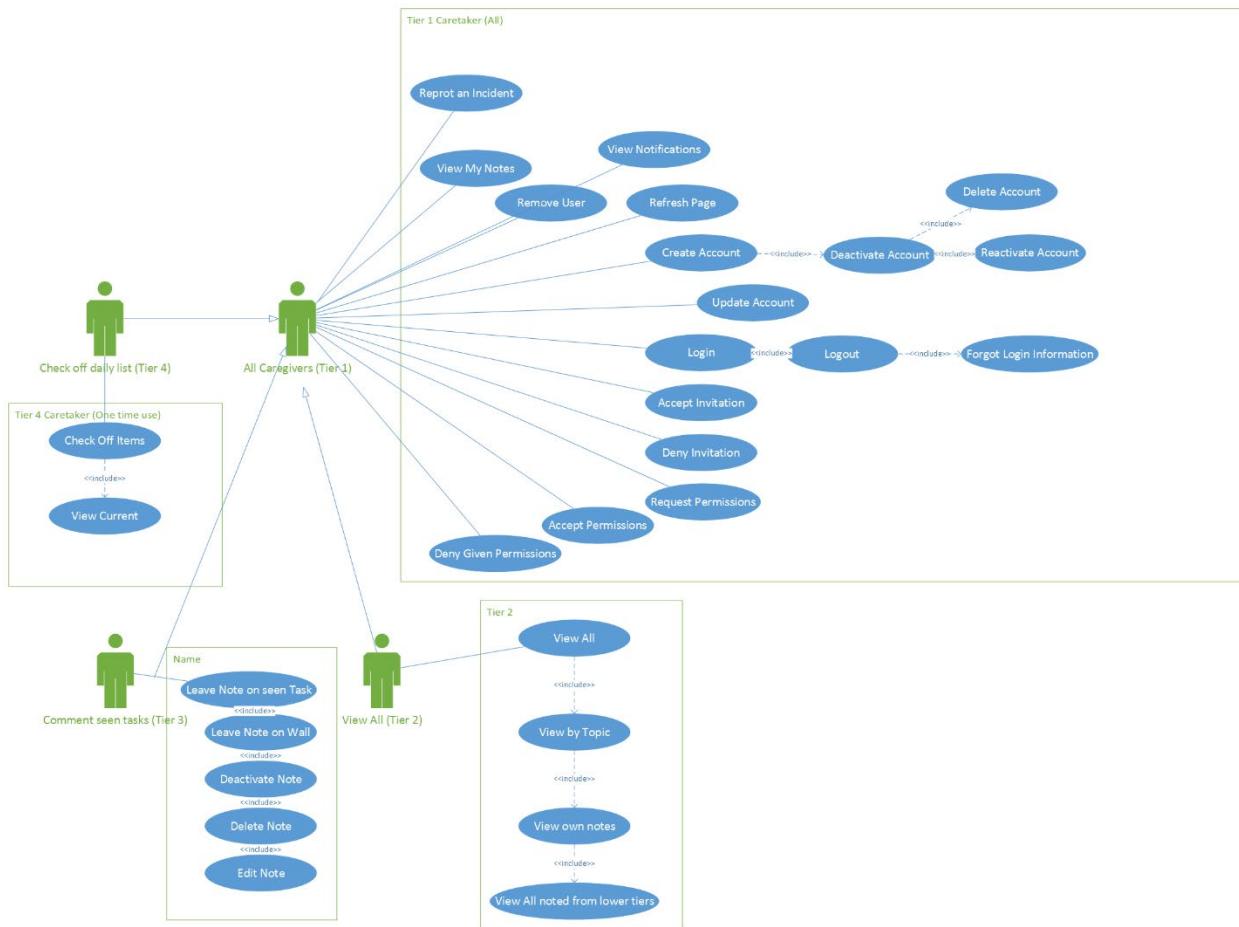
Client Processing Space

By Mitchell Paul



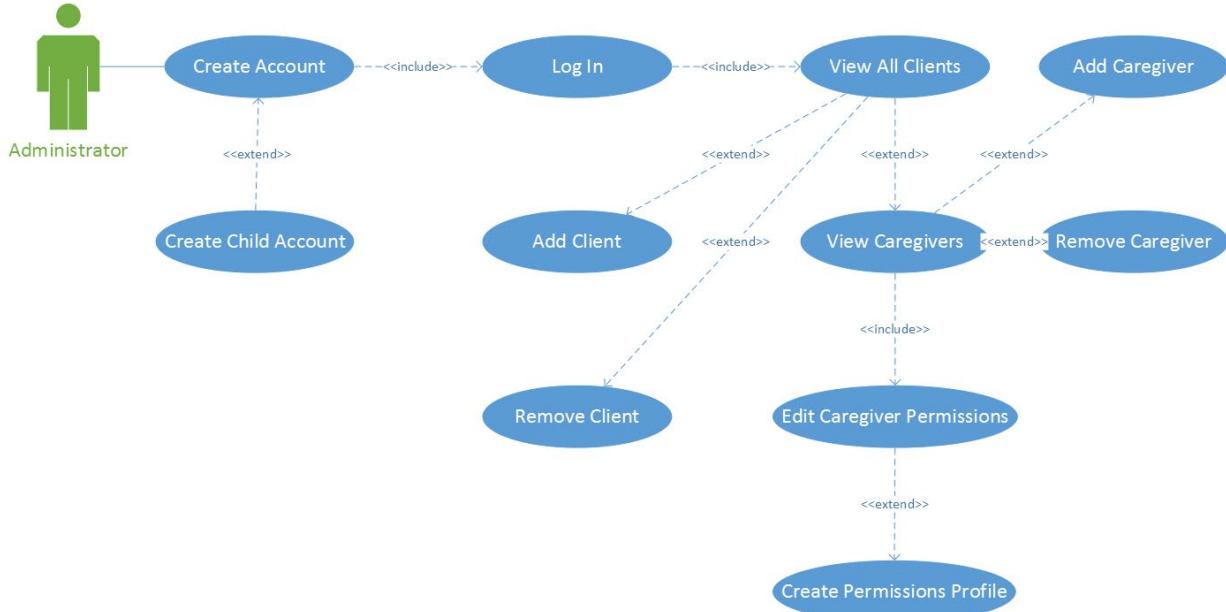
Caregiver Processing Space

By Jory Wernette, contribution by Ryan Taylor



Administrative Processing Space

By William Clark



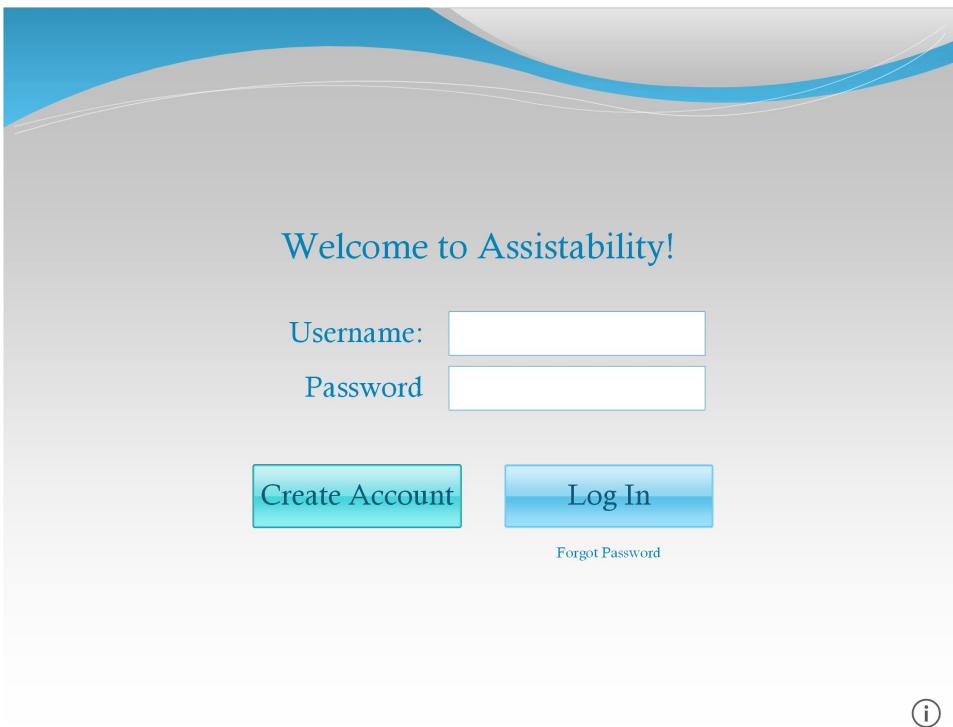
User Interface Drawings

By Jory Wernette, Whitney Vinson

Login and Account Creation Interfaces

By Whitney Vinson

Mobile Landscape



My Account Information

First Name: Johnny

Last Name: Test

Username: JTest

Password: password1

Re-enter Password: password1

Email: email@email.email

Add Picture:



Back

Continue



Select Account Type:

- [Build My Own Goals](#)
- [Build My Child's Goals](#)
- [I am a Caregiver](#)



Back

Continue



My Child's Information

First Name: Johnny
Last Name: Test
Username: JTest
Password: password12
Re-enter Password: password12
Birthdate: 03/25/2010

Add Picture:



Back

Continue



Enter Your Buddy's Information

First Name: Victor
Last Name: Smith
Username: VSmith
Password: password1812
Re-enter Password: password1812
Birthdate: 12/19/1944

Add Picture:



Back

Continue



Select Diagnosis:

Cognitive

(i)

Behavioral

(i)

Physical

(i)

Search for Diagnosis



Skip this Step

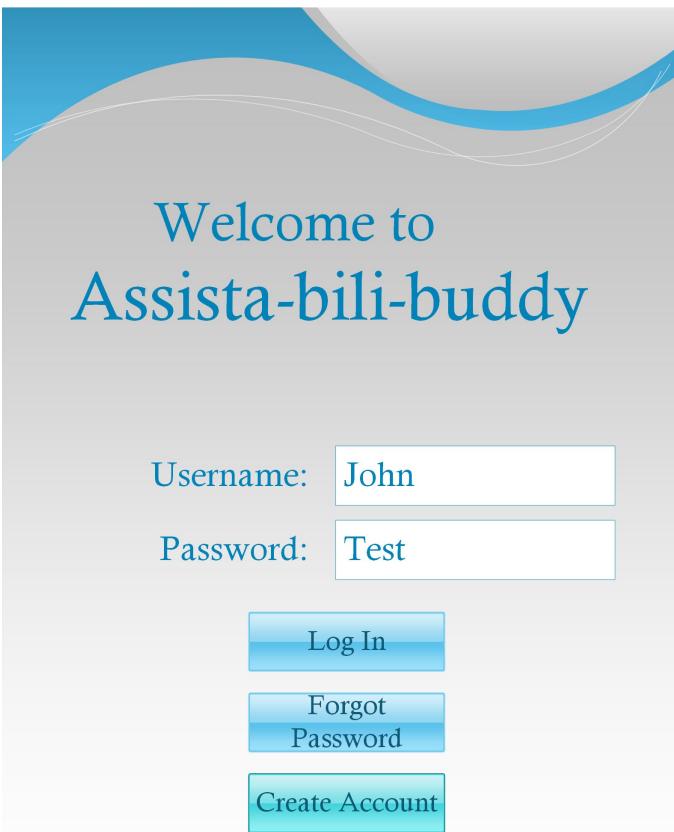
Continue

(i)



Back

Mobile Portrait



My Account Information

First Name: John

Last Name: Test

Username: JTest

Password: password1

Re-enter Password: password1

Birthdate: 10/23/1978

Add Picture:



Continue



Select Account Type:

Build My Own Goals (i)

Build My Child's Goals (i)

I am a Caregiver (i)



Continue

My Child's Information

First Name: John

Last Name: Test

Username: JTest

Password: password1

Re-enter Password: password1

Birthdate: 10/23/1978

Add Picture:



Continue

Enter Your Buddy's Information

First Name: Victor

Last Name: Smith

Username: VSmith

Password: password1812

Re-enter Password: password1812

Birthdate: 12/19/1944

Add Picture:



Continue

Select Diagnosis:

Cognative 

Behavioral 

Physical 

Search for Diagnosis 



Continue

Caregiver Information

First Name:

Last Name:

Username:

Password:

Re-enter Password:

Title:

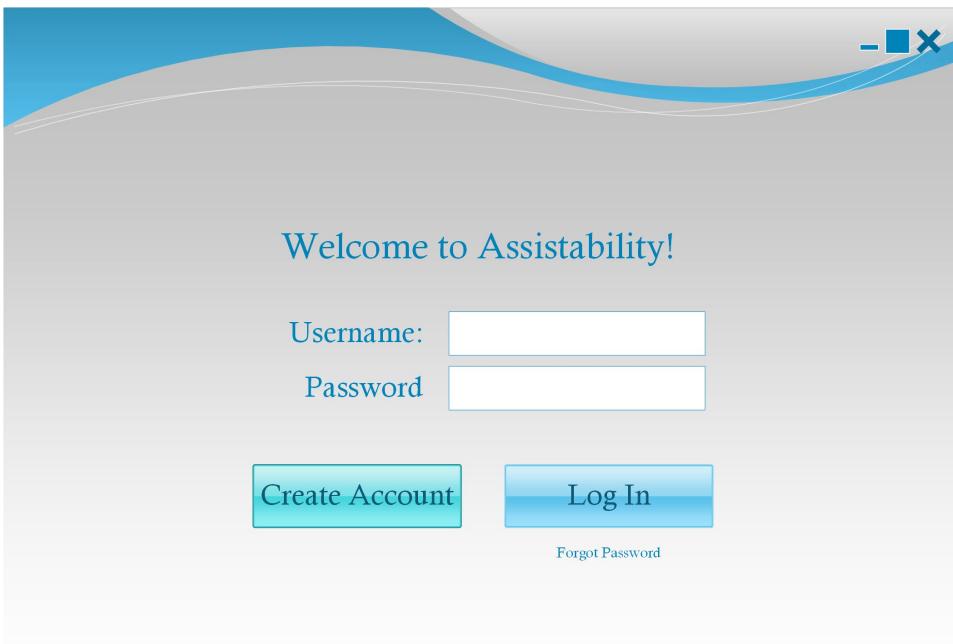
Add Picture:



[Continue](#)



Desktop



Welcome to Assistability!

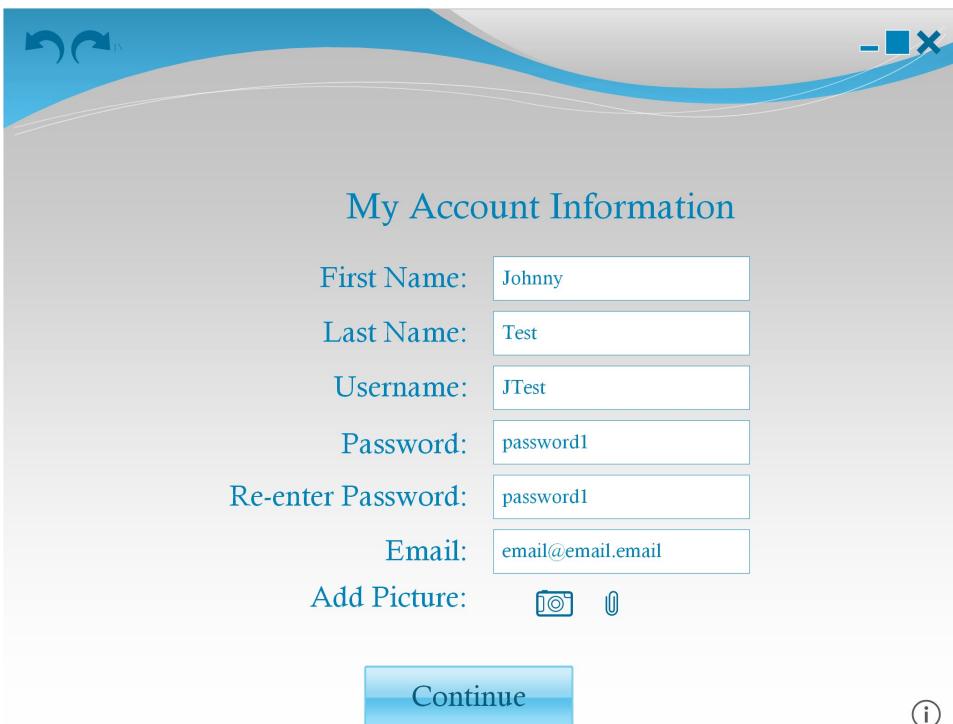
Username:

Password:

[Create Account](#) [Log In](#)

[Forgot Password](#)

(i)



My Account Information

First Name:

Last Name:

Username:

Password:

Re-enter Password:

Email:

Add Picture:

[Continue](#)

(i)

Select Account Type:

Build My Own Goals (i)

Build My Child's Goals (i)

I am a Caregiver (i)

Continue (i)

My Child's Information

First Name:

Last Name:

Username:

Password:

Re-enter Password:

Birthdate:

Add Picture: camera icon trash icon

Continue (i)

Enter Your Buddy's Information

First Name: Victor

Last Name: Smith

Username: VSmith

Password: password1812

Re-enter Password: password1812

Birthdate: 12/19/1944

Add Picture:

[Continue](#)

Select Diagnosis:

Cognitive

Behavioral

Physical

Search for Diagnosis

[Skip this Step](#) [Continue](#)

Web

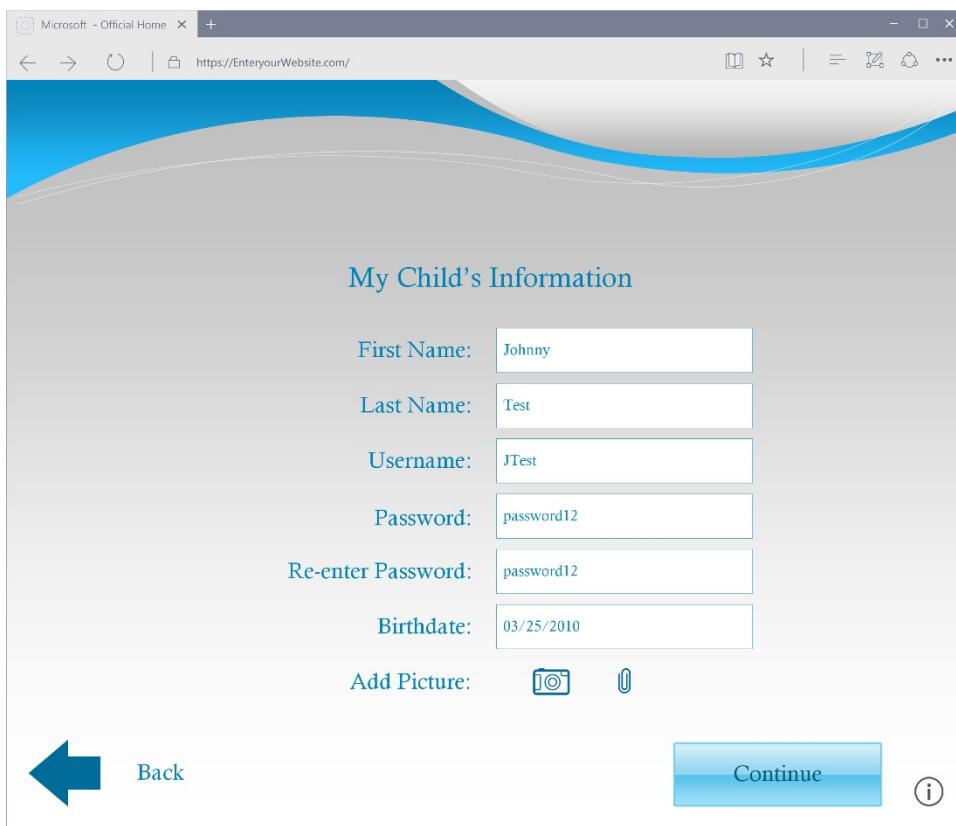
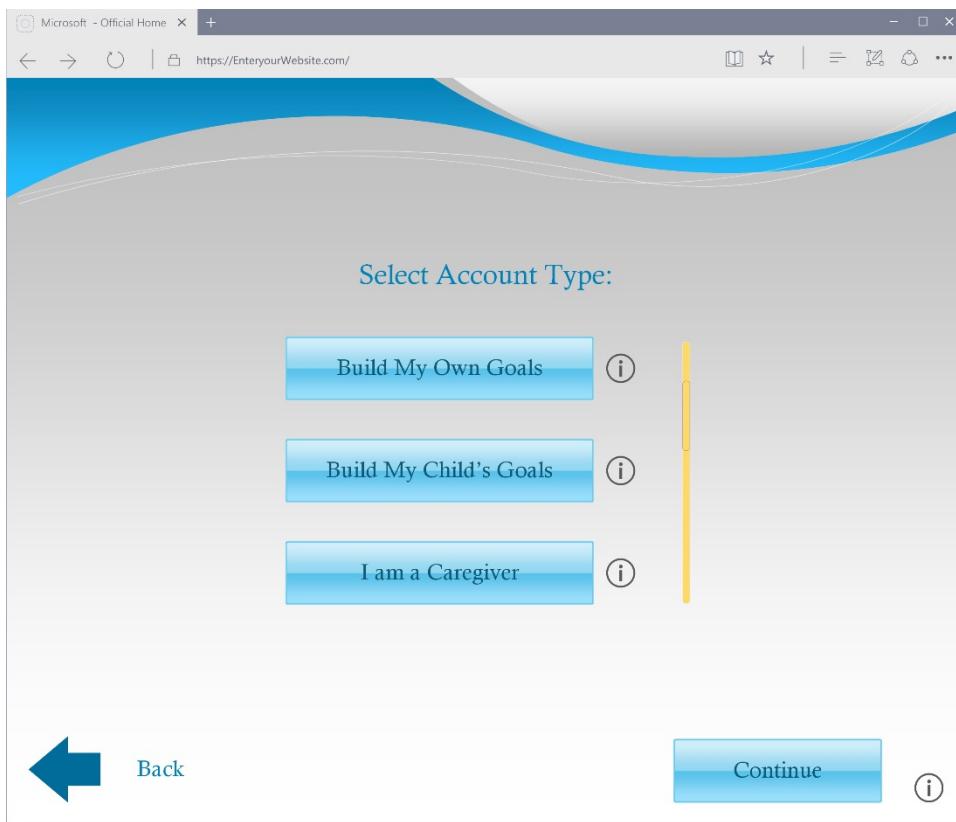
The image displays two screenshots of a web application interface, likely a login page and an account information page, presented in a Microsoft Edge browser window.

Login Page (Top Screenshot):

- Header:** Microsoft - Official Home | https://EnteryourWebsite.com/
- Text:** Welcome to Assistability!
- Form Fields:**
 - Username: [Input Field]
 - Password: [Input Field]
- Buttons:**
 - Create Account
 - Log In
- Links:** Forgot Password
- Information Icon:** ⓘ

Account Information Page (Bottom Screenshot):

- Header:** Microsoft - Official Home | https://EnteryourWebsite.com/
- Text:** My Account Information
- Form Fields:**
 - First Name: Johnny
 - Last Name: Test
 - Username: JTest
 - Password: password1
 - Re-enter Password: password1
 - Email: email@email.email
- Text:** Add Picture:
- Buttons:**
 - Back
 - Continue
- Information Icon:** ⓘ



Microsoft - Official Home

https://EnteryourWebsite.com/

Enter Your Buddy's Information

First Name: Victor

Last Name: Smith

Username: VSmith

Password: password1812

Re-enter Password: password1812

Birthdate: 12/19/1944

Add Picture:

Back Continue

Microsoft - Official Home

https://EnteryourWebsite.com/

Select Diagnosis:

Cognitive

Behavioral

Physical

Search for Diagnosis

Skip this Step

Back Continue

Microsoft - Official Home x +

https://EnteryourWebsite.com/

Caregiver Invitation

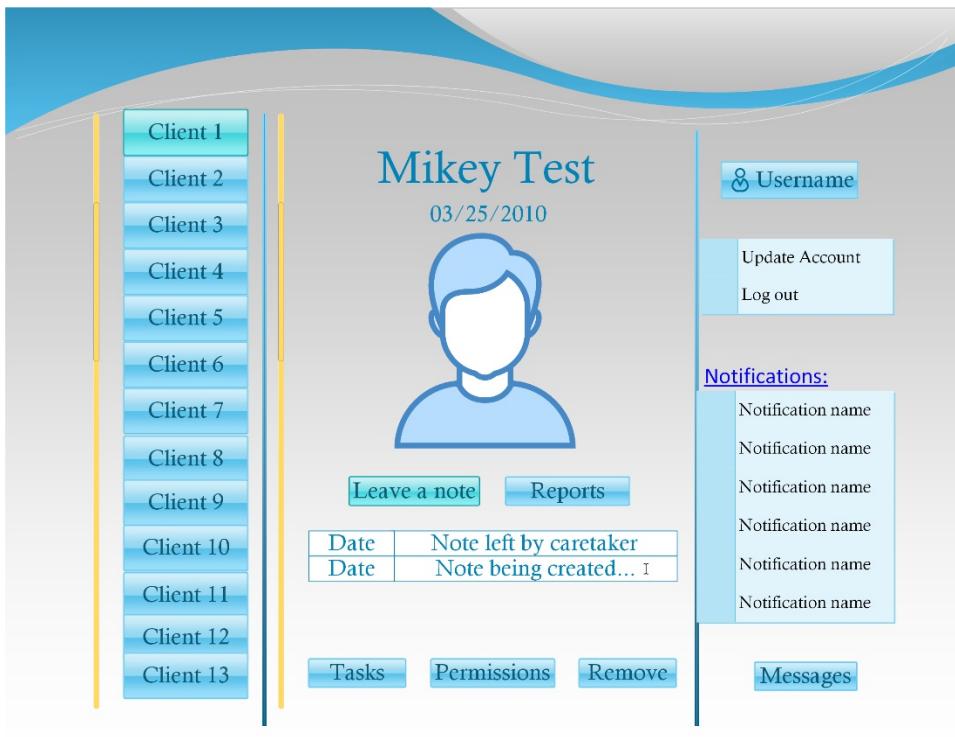
First Name:	Mary
Last Name:	Poppins
Username:	MPoppins
Password:	Sup3rcal1frag1ll1st1c!
Re-enter Password:	Sup3rcal1frag1ll1st1c!
Title:	Nanny

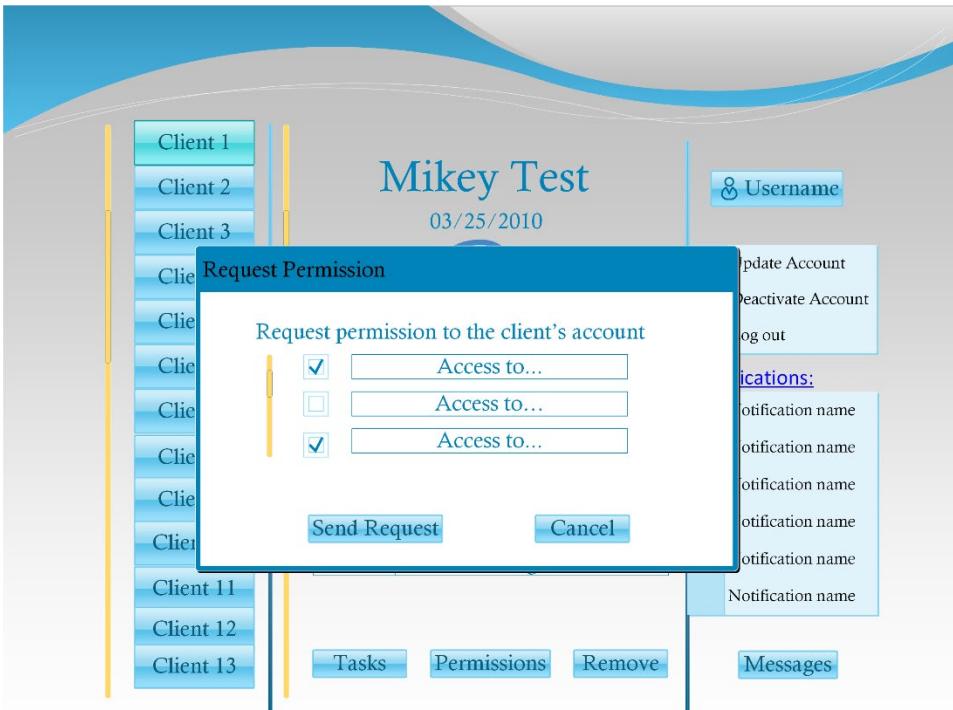
 Back Continue 

Caregiver Interfaces

By Jory Wernette and Ryan Taylor

Mobile Landscape





Johnny Test: Reports

Sort By: Seen Date Name

Username:

Client 1	Date	Name
Client 2	Date	Name
Client 3	Date	Name
Client 4	Date	Name
Client 5	Date	Name
Client 6	Date	Name
Client 7	Date	Name
Client 8	Date	Name
Client 9	Date	Name
Client 10	Date	Name
Client 11	Date	Name
Client 12	Date	Name
Client 13	Date	Name

Incident Report

Date/Time:

Persons Involved:

Description:

Repercusion:

Actions:

- [Logout](#)
- [Activate Account](#)
- [Notifications](#)
- [Messages](#)

Submit **Cancel**

Johnny Test: Daily Tasks:

Sort By: [Done](#) [Date](#) [Name](#)

	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name

[Refresh](#) [Remove Client](#) [All Tasks](#)

[Messages](#)

& Username

- [Update Account](#)
- [Deactivate Account](#)
- [Log out](#)

Notifications:

- Notification name

Deactivating account

Deactivation warning

Username:

Password :

[Deactivate](#)

[Cancel](#)

Removing Client

Removing Client warning

Username:

Password :

[Remove Client](#)

[Cancel](#)

Permission

Permission being given description

[Accept](#)

[Cancel](#)

[Deny](#)

All Notifications:

Sort By: [Read](#) [Date](#) [Name](#) [Description](#)

<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.

[Refresh](#)

Username
[Update Account](#)
[Deactivate Account](#)
[Log out](#)

Notifications:

- [Notification name](#)

Messages

All Notifications:

Sort By: [Read](#) [Date](#) [Name](#) [Description](#)

<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.

[Accept](#) [Decline](#)

Client Invitation

Would you like to be linked to
"Client's Name"
Account?

Your Access: Tier 4 Caretaker
(You will be allowed to see the client's daily tasks and mark them as complete)

Username
[Update Account](#)
[Deactivate Account](#)

Notifications:

- [Notification name](#)

Messages

Update Account Information


Take New
Upload

First Name: John
 Last Name: Test
 Username: JTest
 Password: password1
 Re-enter Password: password1

Save **Cancel**

My Notes

& Username

- Update Account
- Deactivate Account
- Log out

Notifications:

- Notification name

Messages

Client Name Task:

Done **Date** **Name**

Date Name

Task Description...

Leave a note

Date Note being created...

Refresh **Remove Client** **All Tasks**

& Username

- Update Account
- Deactivate Account
- Log out

Notifications:

- Notification name

Messages

Your Notes

Date	Location	Content
Date	Location	Contents
Date	Task	Contents
Date	Wall	Contents

Client List:

- Client 1
- Client 2
- Client 3
- Client 4
- Client 5
- Client 6
- Client 7
- Client 8
- Client 9
- Client 10
- Client 11
- Client 12
- Client 13

Actions:

- Edit
- Delete

Username:

- Update Account
- Deactivate Account
- Log out

Notifications:

- Notification name

Messages

Messages

Users



Admins



Message Content:

I noticed Johnny has been doing better finishing his homework. I really appreciate your help!

Johnny is such a good kid, I am so glad I can be a part of helping him!

Actions:

- Send

Username:

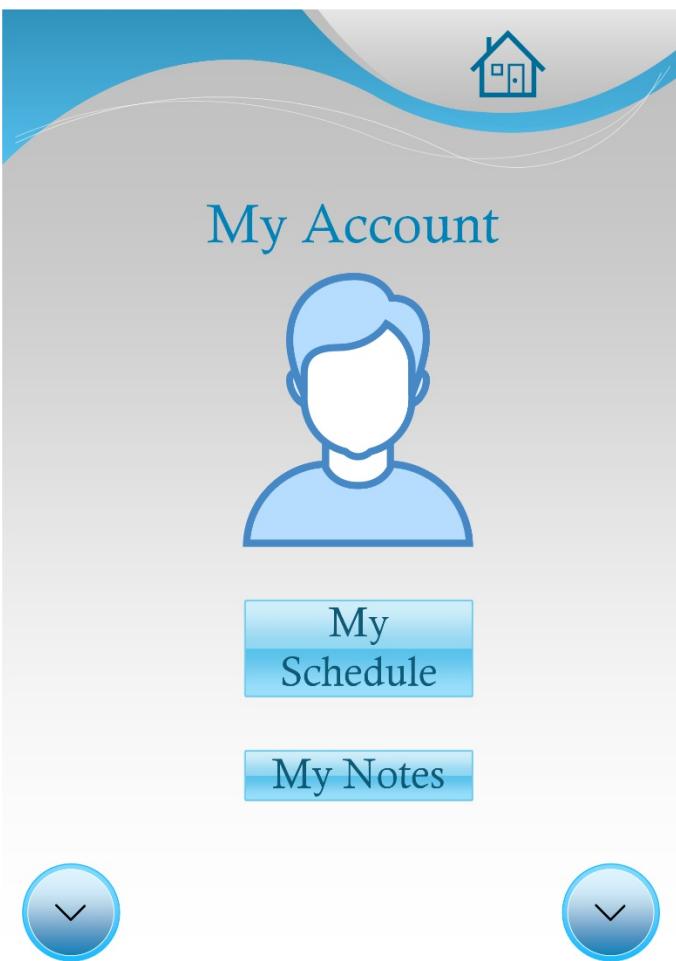
- Update Account
- Deactivate Account
- Log out

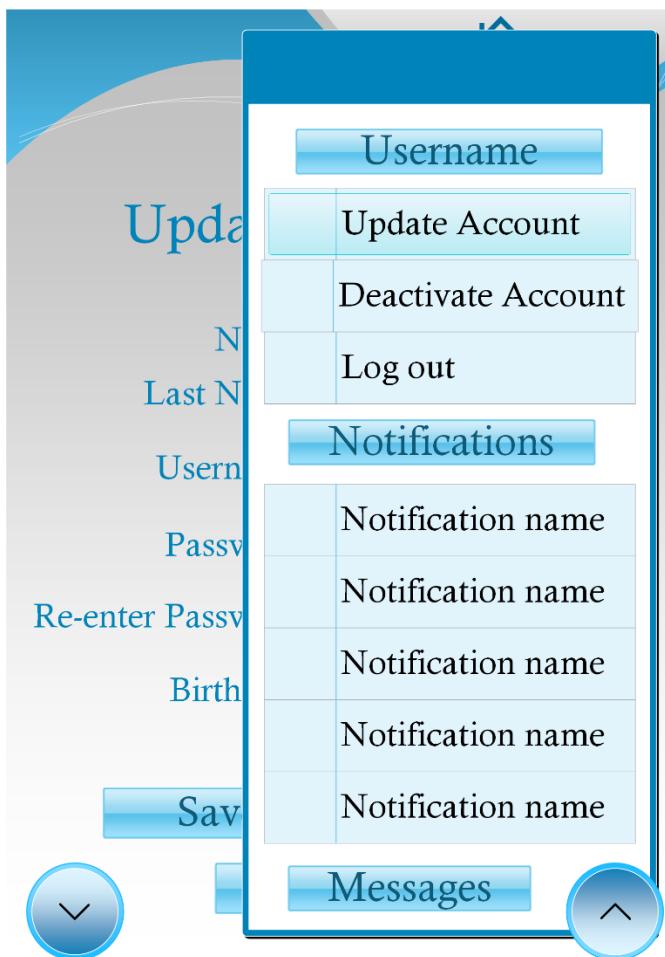
Notifications:

- Notification name

Messages

Mobile Portrait







Update Account



Take New



Upload

First Name: John

Last Name: Test

Username: JTest

Password: Password!1

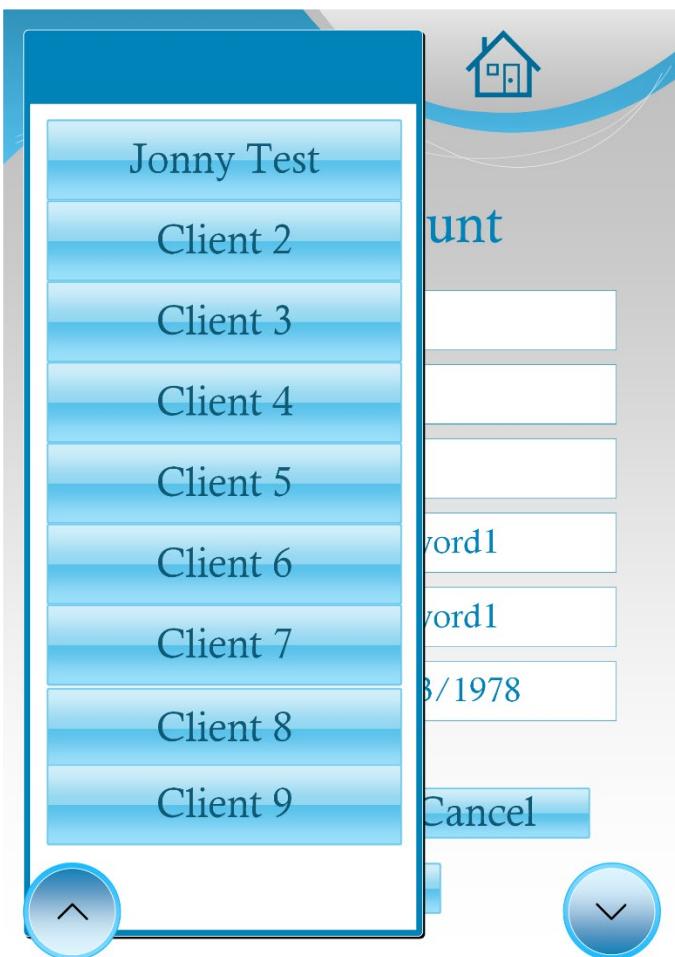
Save

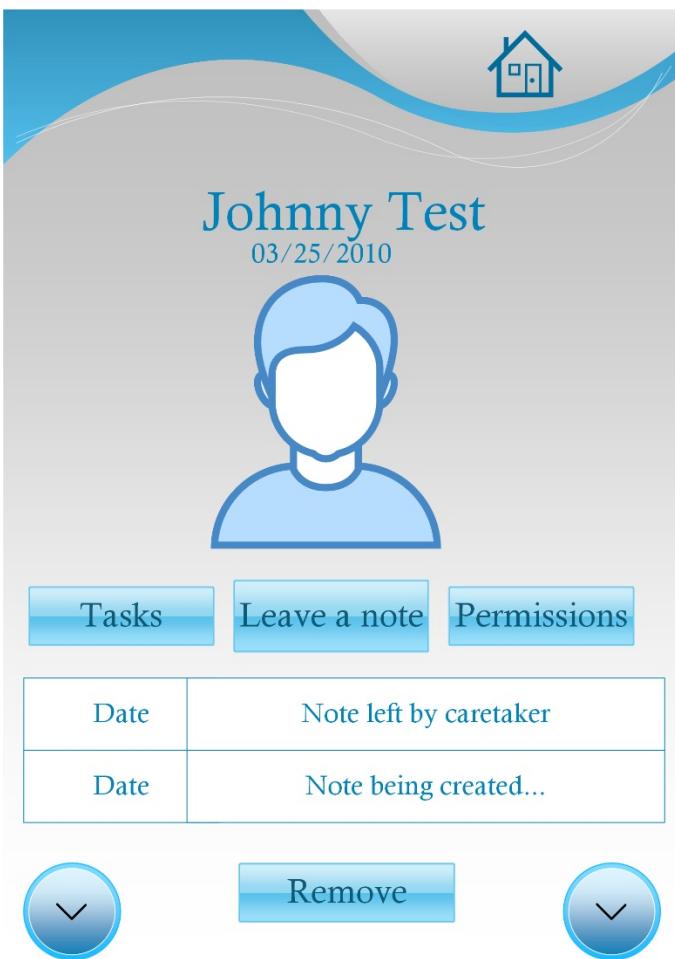
Cancel



My Notes



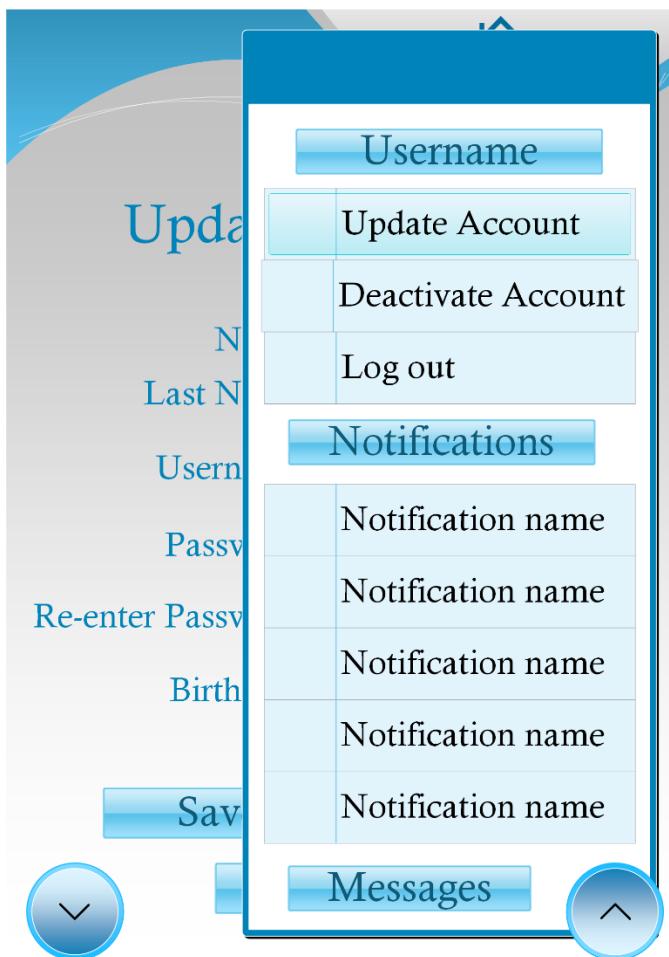




A user profile interface for "Johnny Test" dated 03/25/2010. The profile picture is a blue silhouette of a young boy's head and shoulders. Below the profile are three buttons: "Tasks", "Leave a note", and "Permissions". A table follows, showing two rows of notes. The first row has "Date" and "Note left by caretaker". The second row has "Date" and "Note being created...". At the bottom are two circular arrows pointing down.

Date	Note left by caretaker
Date	Note being created...

Tasks Leave a note Permissions



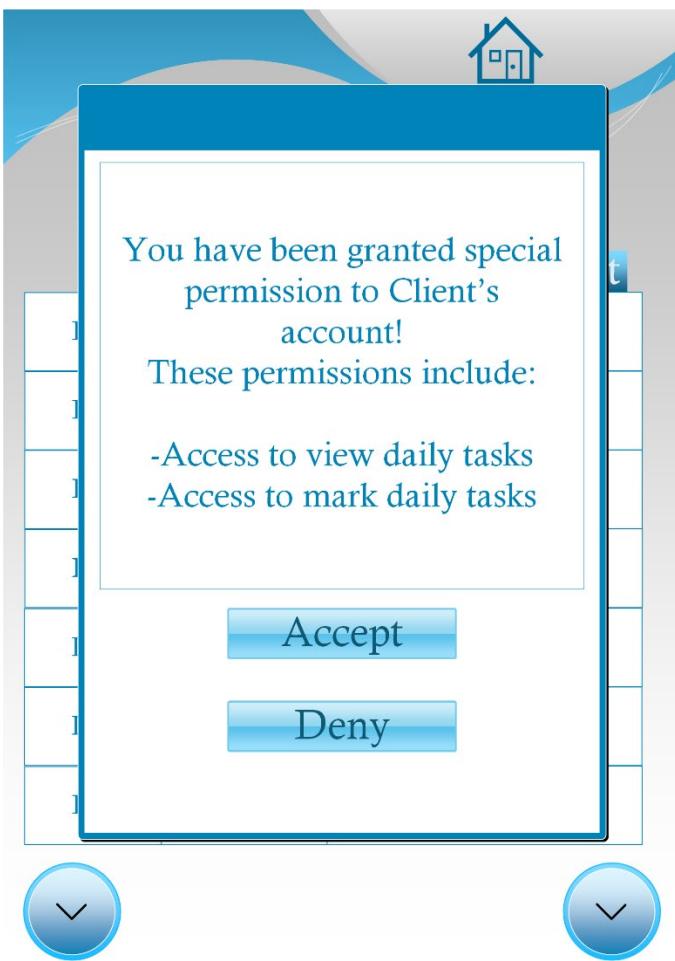


Messages

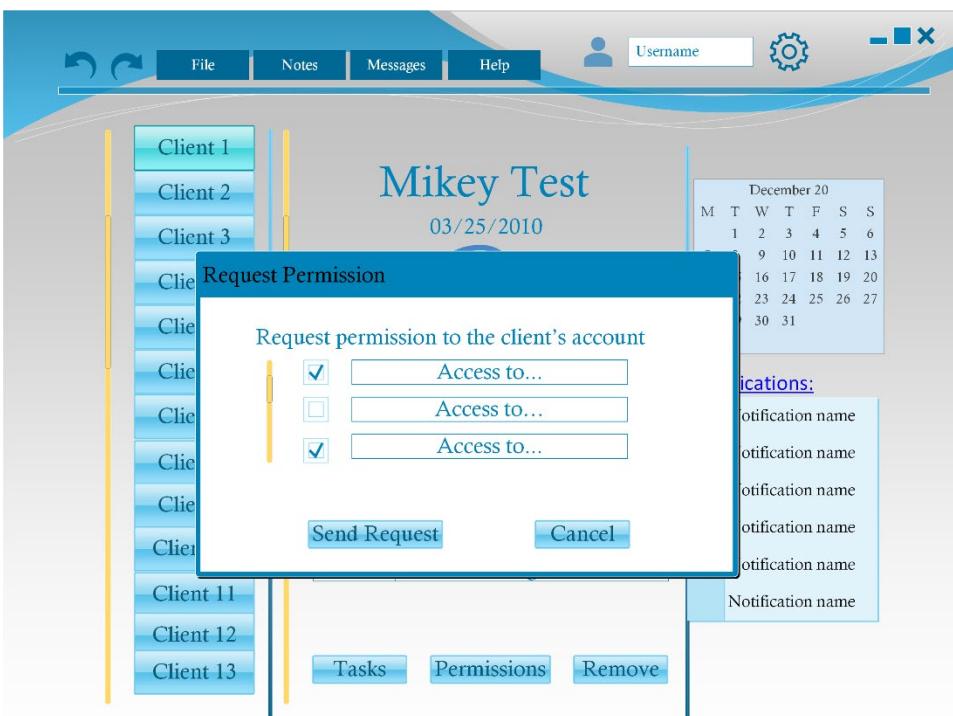
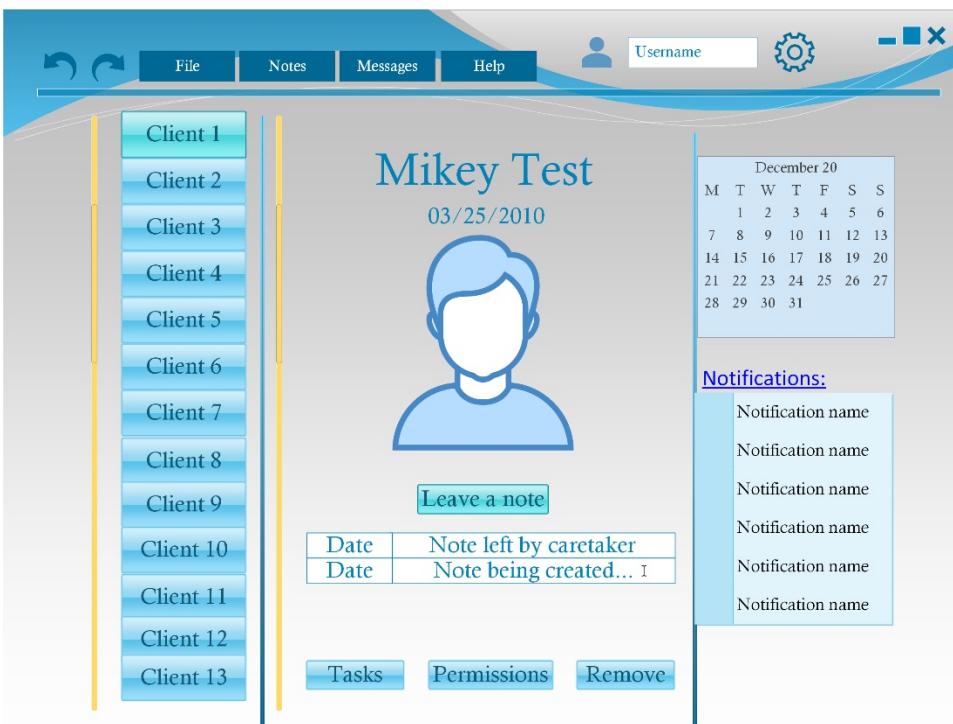
Sort

Date	Name	Desc.

◀ ▶



Desktop



The screenshot shows the software's main window with a blue header bar containing 'File', 'Notes', 'Messages', 'Help', a user icon, 'Username', and a gear icon. Below the header is a sidebar with a vertical scroll bar on the right, listing clients from 'Client 1' to 'Client 13'. The main content area is titled 'Johnny Test: All Tasks' and displays a table of tasks. The table has columns for 'Done' (with a checked checkbox), 'Date' (containing 'Date'), and 'Name' (containing 'Name'). A 'Sort By' dropdown menu is set to 'Done'. To the right of the table is a calendar for December 2020 and a 'Notifications:' section listing six entries. At the bottom are buttons for 'Refresh', 'Remove Client', and 'Daily Tasks'.

	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name

This screenshot shows the software's interface with the 'Reports' section active. The header and sidebar are identical to the first screenshot. The main content area is titled 'Johnny Test: Reports' and features a 'Sort By' dropdown set to 'Seen'. An 'Incident Report' dialog box is open in the center, containing fields for 'Date/Time' (12/01/20), 'Persons Involved' (Donnie + Leo), 'Description' (Johnny threw lamp at Donnie and Leo then said lighten up), and 'Repercussion' (No lamps for a week). Below the dialog are buttons for 'Submit' and 'Cancel'. To the right of the dialog is a calendar for December 2020 and a 'Notifications:' section. At the bottom are buttons for 'Refresh', 'Add', and 'Remove'.

	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name

Johnny Test: Daily Tasks:

Son By: [Done](#) [Date](#) [Name](#)

✓	Date	Name
✓	Date	Name

December 20
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Notifications:

- Notification name

[Refresh](#) [Remove Client](#) [All Tasks](#)

Deactivating account

Deactivation warning

Username:

Password :

[Deactivate](#) [Cancel](#)

Removing Client

Removing Client warning

Username:

Password :

[Remove Client](#) [Cancel](#)

Permission

Permission being given description

[Accept](#) [Cancel](#) [Deny](#)

The screenshot shows a software interface with a header containing 'File', 'Notes', 'Messages', 'Help', 'Username', and a gear icon. Below the header is a sidebar with a list of clients: Client 1, Client 2, Client 3, Client 4, Client 5, Client 6, Client 7, Client 8, Client 9, Client 10, Client 11, Client 12, and Client 13. The main area is titled 'All Notifications:' and displays a table of notifications. The table has columns: 'Read' (checkbox), 'Date', 'Name', and 'Desc.'. There are 15 rows, each with a checked 'Read' checkbox and the same date ('Date') and description ('Desc.') values. Below the table is a 'Refresh' button. To the right of the table is a calendar for December 20, showing days from 1 to 31. Further right is a section titled 'Notifications:' containing six entries: 'Notification name', 'Notification name', 'Notification name', 'Notification name', 'Notification name', and 'Notification name'.

This screenshot is similar to the one above, showing the 'All Notifications:' screen. However, a modal dialog box is overlaid on the client list. The dialog is titled 'Client Invitation' and contains the message: 'Would you like to be linked to "Client's Name" Account?'. It also includes the text 'Your Access: Tier 4 Caretaker' and '(You will be allowed to see the client's daily tasks and mark them as complete)'. At the bottom of the dialog are two buttons: 'Accept' and 'Decline'. The rest of the interface, including the calendar and notification section, remains visible.

File Notes Messages Help Username    

Client 1
Client 2
Client 3
Client 4
Client 5
Client 6
Client 7
Client 8
Client 9
Client 10
Client 11
Client 12
Client 13

Update Account Information



 Take New
 Upload

First Name: John
Last Name: Test
Username: JTest
Password: password1
Re-enter Password: password1

Save **Cancel** **My Notes**

December 20
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Notifications:
Notification name
Notification name
Notification name
Notification name
Notification name
Notification name

File Notes Messages Help Username    

Client 1
Client 2
Client 3
Client 4
Client 5
Client 6
Client 7
Client 8
Client 9
Client 10
Client 11
Client 12
Client 13

Client Name Task:

Done **Date** **Name**

Date Name

Task Description...

Leave a note

Date Note being created...

Refresh **Remove Client** **All Tasks**

December 20
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Notifications:
Notification name
Notification name
Notification name
Notification name
Notification name
Notification name

File Notes Messages Help

Username

Client 1 Client 2 Client 3 Client 4 Client 5 Client 6 Client 7 Client 8 Client 9 Client 10 Client 11 Client 12 Client 13

Your Notes

Date	Location	Content
Date	Location	Contents
Date	Task	Contents
Date	Wall	Contents

December 20

M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

- Notification name

Edit Delete

File Notes Messages Help

Username

Client 1 Client 2 Client 3 Client 4 Client 5 Client 6 Client 7 Client 8 Client 9 Client 10 Client 11 Client 12 Client 13

Messages

Users

Admins

I noticed Johnny has been doing better finishing his homework. I really appreciate your help!

Johnny is such a good kid, I am so glad I can be a part of helping him!

Send

December 20

M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

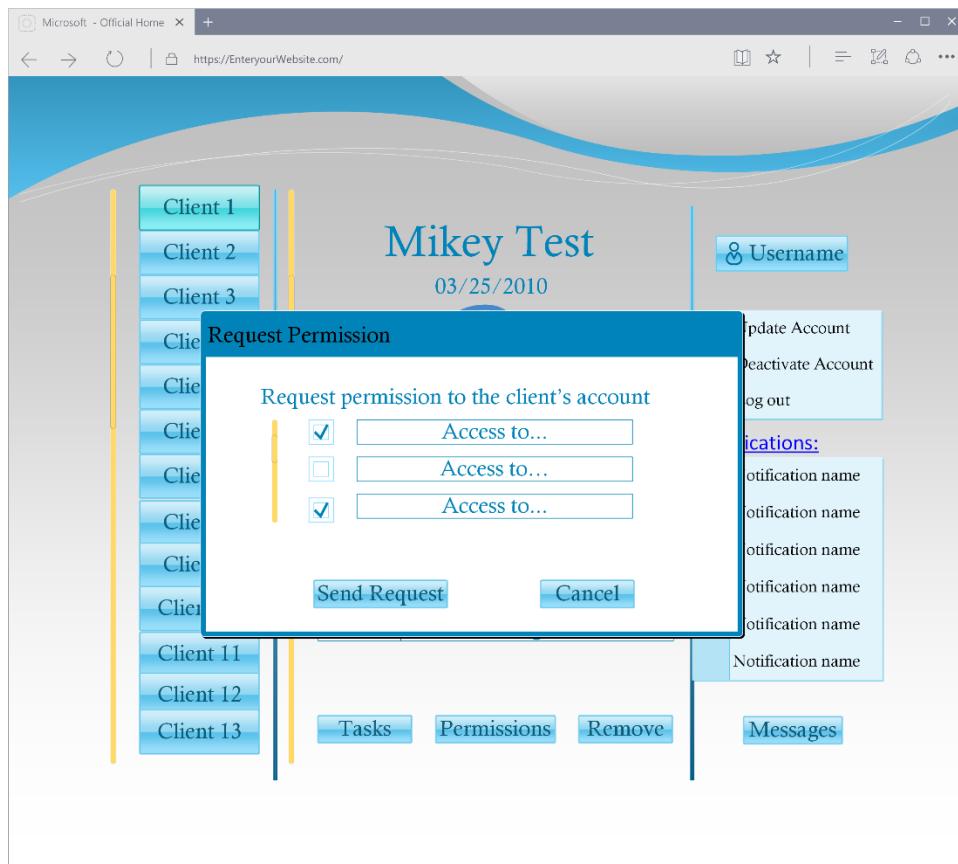
- Notification name

Web

The screenshot shows a web browser window with a light blue header bar. The address bar displays "https://EnteryourWebsite.com/". The main content area has a light gray background with a blue decorative swoosh at the top. On the left, there is a vertical sidebar with a list of clients from "Client 1" to "Client 13". In the center, a user profile for "Mikey Test" is shown, with the date "03/25/2010" below it. A blue placeholder icon of a person's head and shoulders is displayed. Below the profile are two buttons: "Leave a note" and "Reports". Underneath these buttons is a table with two rows:

Date	Note left by caretaker
Date	Note being created... 1

Below the table are three buttons: "Tasks", "Permissions", and "Remove". To the right of the central area is a sidebar with a light blue header labeled "Notifications:". This sidebar contains a list of six notifications, each labeled "Notification name". At the bottom of the sidebar is a button labeled "Messages". On the far right of the page, there is a vertical sidebar with a light blue header labeled "Username". It contains two buttons: "Update Account" and "Log out".



Microsoft - Official Home https://EnterYourWebsite.com/

Johnny Test: All Tasks

Sort By: Done Date Name

	Done	Date	Name
1	<input checked="" type="checkbox"/>	Date	Name
2	<input checked="" type="checkbox"/>	Date	Name
3	<input checked="" type="checkbox"/>	Date	Name
4	<input checked="" type="checkbox"/>	Date	Name
5	<input checked="" type="checkbox"/>	Date	Name
6	<input checked="" type="checkbox"/>	Date	Name
7	<input checked="" type="checkbox"/>	Date	Name
8	<input checked="" type="checkbox"/>	Date	Name
9	<input checked="" type="checkbox"/>	Date	Name
10	<input checked="" type="checkbox"/>	Date	Name
11	<input checked="" type="checkbox"/>	Date	Name
12	<input checked="" type="checkbox"/>	Date	Name
13	<input checked="" type="checkbox"/>	Date	Name

Client 1 Client 2 Client 3 Client 4 Client 5 Client 6 Client 7 Client 8 Client 9 Client 10 Client 11 Client 12 Client 13

Refresh Remove Client Daily Tasks

Username

Update Account
Deactivate Account
Log out

Notifications:

Notification name
Notification name
Notification name
Notification name
Notification name
Notification name

Messages

The screenshot displays a web-based application interface. On the left, a vertical list of clients is shown, each with a blue background and white text. In the center, a table lists 13 tasks for 'Client 1'. The table has three columns: 'Done' (with checkboxes), 'Date' (sorted by date), and 'Name' (sorted by name). Below the table are three buttons: 'Refresh', 'Remove Client', and 'Daily Tasks'. To the right, a sidebar contains a 'Username' dropdown menu with 'Update Account', 'Deactivate Account', and 'Log out' options. Under 'Notifications:', there are six entries, each consisting of 'Notification name'. At the bottom right of the sidebar is a 'Messages' link.

Microsoft - Official Home https://EnteryourWebsite.com/

Johnny Test: Daily Tasks

Sort By: [Done](#) [Date](#) [Name](#)

	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name
<input checked="" type="checkbox"/>	Date	Name

[Refresh](#) [Remove Client](#) [All Tasks](#)

[Username](#)

- [Update Account](#)
- [Deactivate Account](#)
- [Log out](#)

Notifications:

- Notification name

[Messages](#)

Microsoft - Official Home + https://EnteryourWebsite.com/

Johnny Test: Reports

Sort By: Seen Date Name

	Date	Name
Client 1		
Client 2	<input checked="" type="checkbox"/>	Date
Client 3	<input checked="" type="checkbox"/>	Name
Client 4		
Client 5		
Client 6		
Client 7		
Client 8		
Client 9		
Client 10		
Client 11	<input checked="" type="checkbox"/>	Date
Client 12		
Client 13		

Username:

Incident Report

Date/Time: 12/01/20

Persons Involved: Donnie + Leo

Description: Johnny threw lamp at Donnie and Leo then said lighten up

Repercusion: No lamps for a week

Submit Cancel

Refresh Add Remove

Notification name

Messages

Microsoft - Official Home https://EnterYourWebsite.com/

All Notifications:

Sort By: [Read](#) [Date](#) [Name](#) [Description](#)

	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.
<input checked="" type="checkbox"/>	Date	Name	Desc.

Refresh

Username

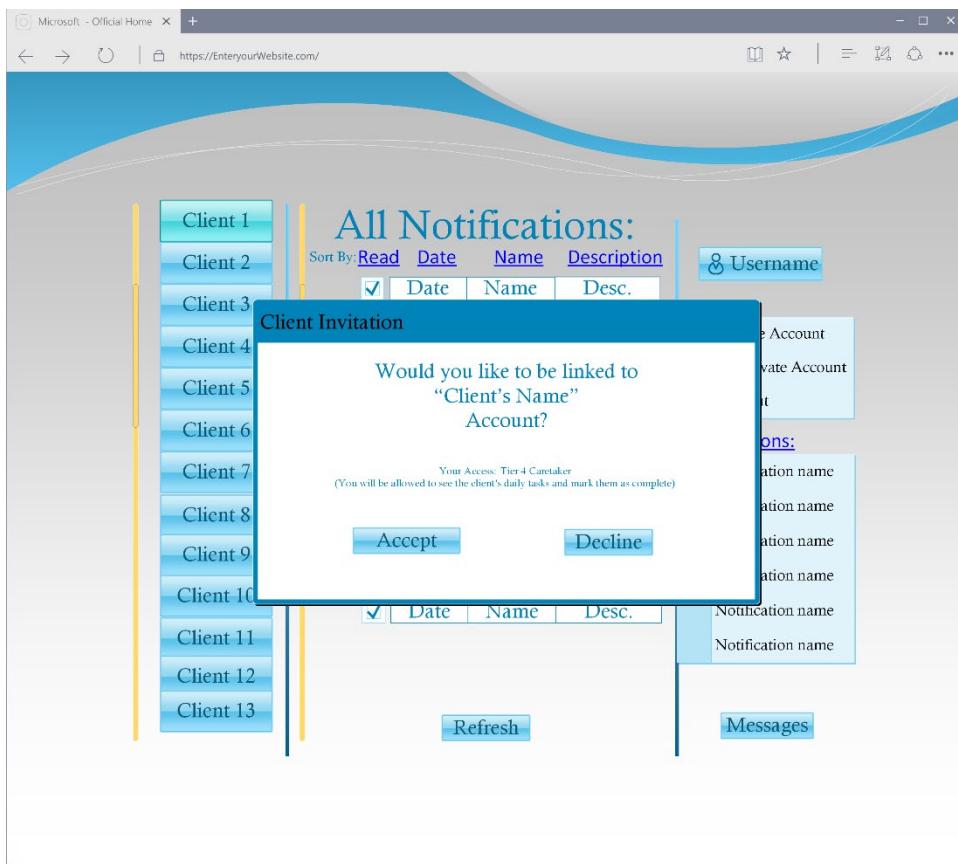
Update Account
Deactivate Account
Log out

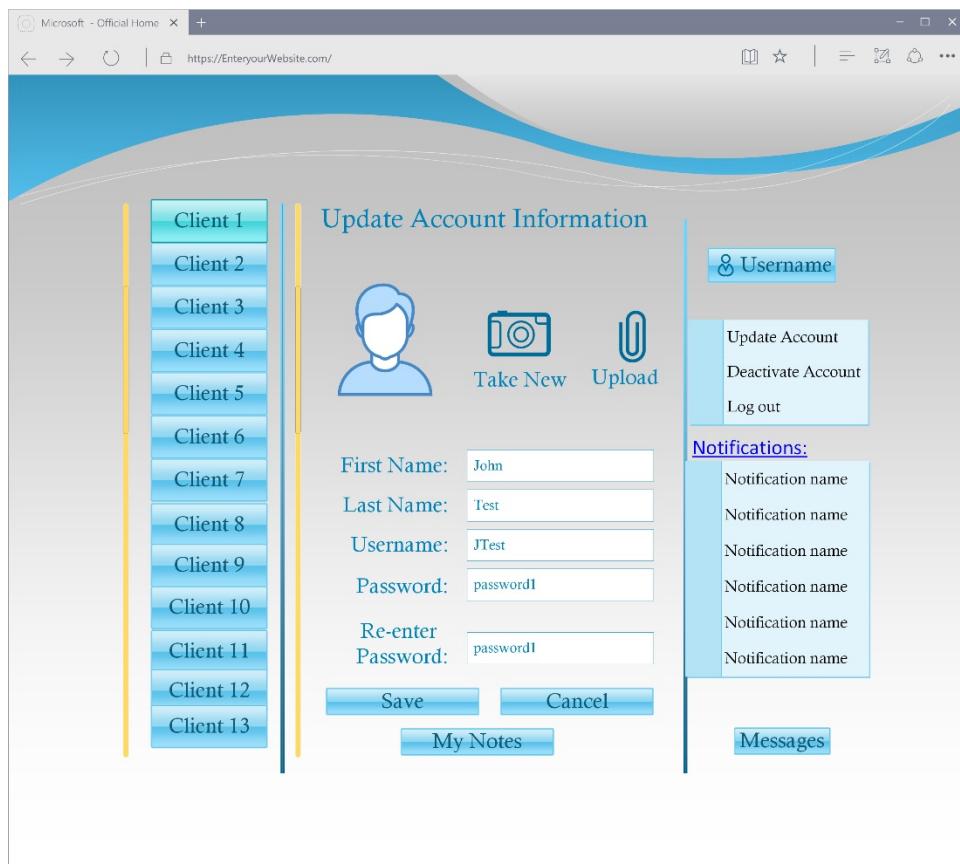
Notifications:

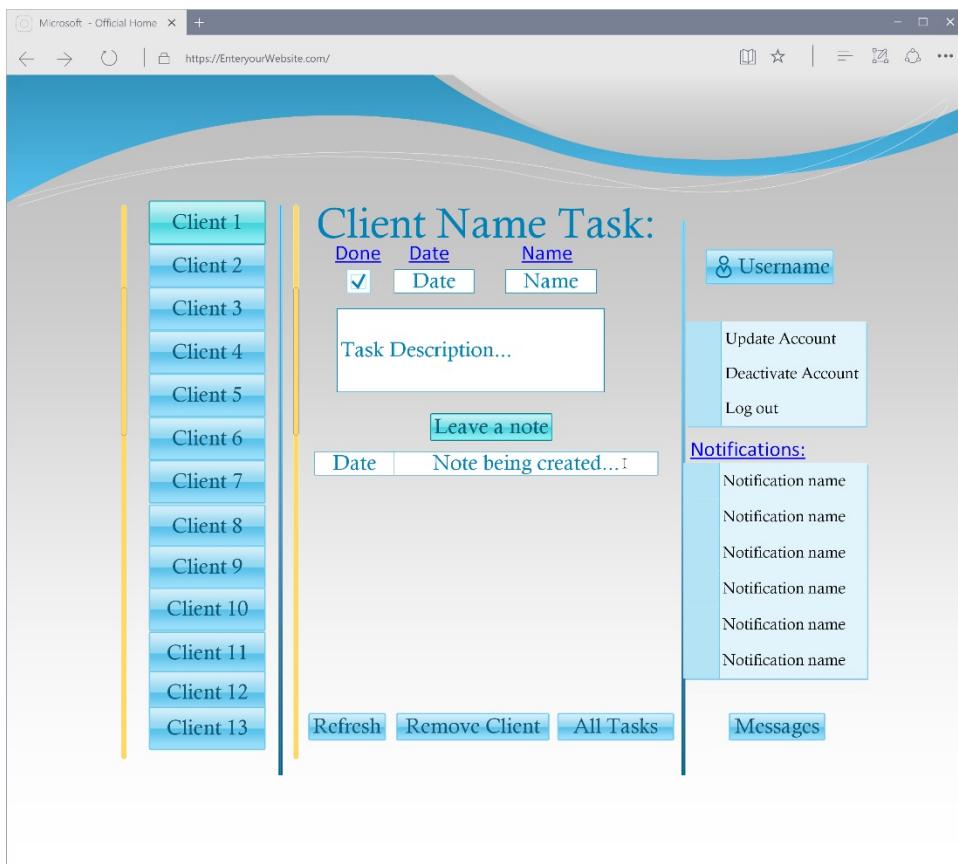
Notification name
Notification name
Notification name
Notification name
Notification name
Notification name

Messages

The screenshot shows a Microsoft Edge browser window with a light gray header bar containing the title 'Microsoft - Official Home', a back/forward button, a refresh button, and the URL 'https://EnterYourWebsite.com/'. Below the header is a decorative blue and white wavy graphic. The main content area has a light gray background. On the left, there's a vertical list of 13 clients labeled 'Client 1' through 'Client 13'. To the right of this list is a large central panel titled 'All Notifications:' which contains a table of notifications. The table has columns for 'Read' (with checkmarks), 'Date', 'Name', and 'Description'. Below the table is a 'Refresh' button. To the right of the notifications panel is a sidebar with sections for 'Username' (containing account management links like 'Update Account', 'Deactivate Account', and 'Log out'), 'Notifications:' (listing six notification names), and 'Messages' (which is currently empty). The entire interface is styled with a clean, modern look using blue, white, and gray colors.







Microsoft - Official Home + https://EnterYourWebsite.com/

Your Notes

Date	Location	Content
Date	Task	Contents
Date	Wall	Contents

Client 1
Client 2
Client 3
Client 4
Client 5
Client 6
Client 7
Client 8
Client 9
Client 10
Client 11
Client 12
Client 13

Edit Delete

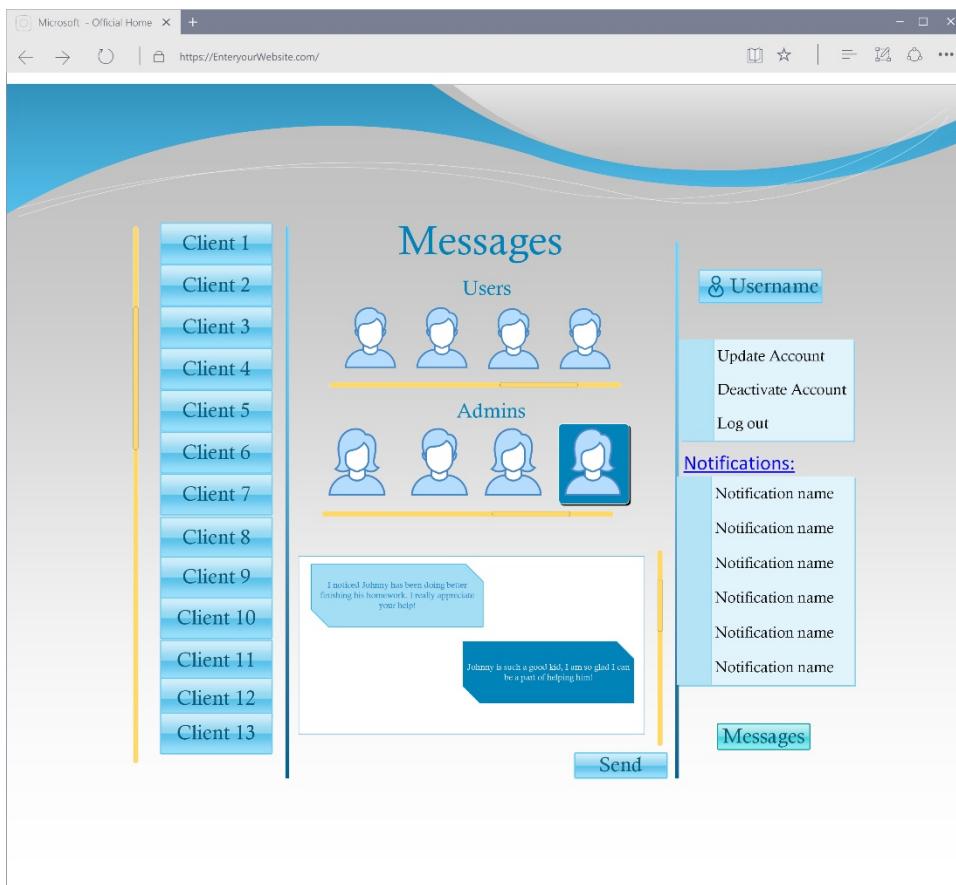
Username

Update Account
Deactivate Account
Log out

Notifications:

Notification name
Notification name
Notification name
Notification name
Notification name
Notification name

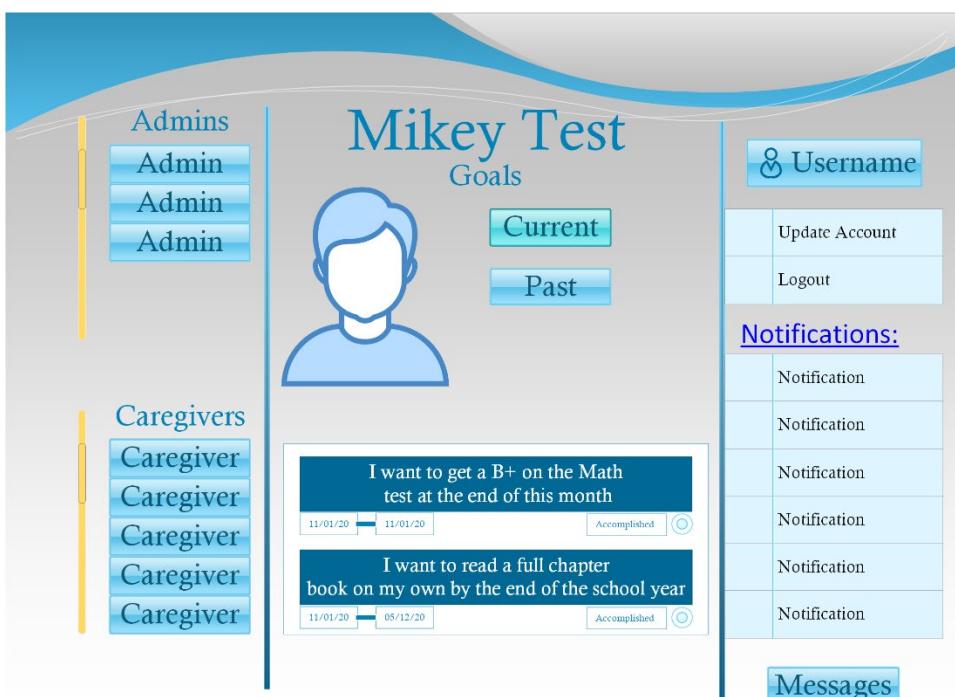
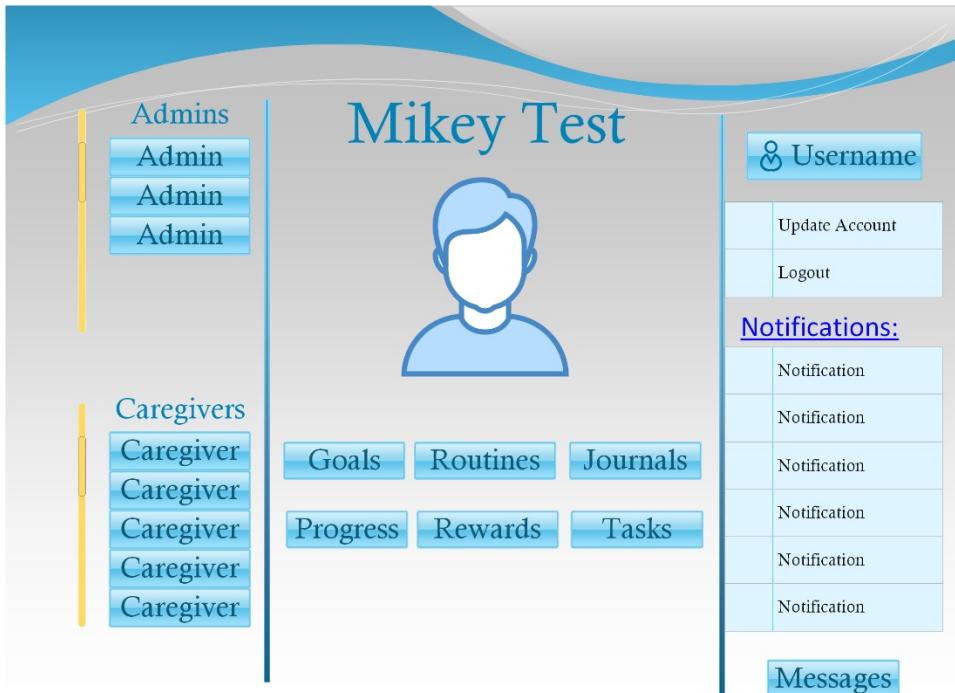
Messages



Client Interfaces

By Jory Wernette, Becky Baenziger, Mitchell Paul, Whitney Vinson, Nathaniel Webber

Mobile Landscape



My Caregivers

Profile	Name
	Mary Poppins Nanny
	Mrs. Doubtfire Nanny
	Ms. Frizzle Teacher
	Dr. Gregory House Doctor

Username

- Update Account
- Logout

Notifications:

Notification

Messages

Ms. Frizzle

Send Message

Username

- Update Account
- Logout

Notifications:

Notification

Messages

Messages

Admins

Caregivers

Send

Messages

Username

Notifications:

Notification

Notification

Notification

Notification

Notification

Notification

Update Account

Logout

You've been doing so great on your tasks! Keep up the good work buddy!

Thanks! I am trying to get to that next reward! I get a new game!

Mikey Test

Progress

Report Cards

History

Username

Notifications:

Notification

Notification

Notification

Notification

Notification

Update Account

Logout

Type

Babysitter Report

School Behavior

School Behavior

School Behavior

Date

11/01/20

10/31/20

10/30/20

10/29/20

10/28/20

Messages

Admins

Admin
Admin
Admin

Caregivers

Caregiver

Mikey Test Journals

Title	Date
Halloween Journal	10/2020
Bikes	09/2020
Mario	08/2020
I like Pokemon	07/2020
What should I go as?	06/2020

[New Journal](#) [View Journal](#)

Username

Update Account
Logout

Notifications:

Notification

[Messages](#)

Admins

Admin
Admin
Admin

Caregivers

Caregiver

Mikey Test Halloween Journal Entries

Title	Date
I ate too much candy	11/01/20
My Halloween	10/31/20
I am so excited!!!	10/30/20
I tried on my new costume!	10/29/20
What should I go as?	10/28/20

[View](#) [Add](#)

Username

Update Account
Logout

Notifications:

Notification

[Messages](#)

The screenshot shows a digital interface for journal entries. At the top, there's a navigation bar with 'Admins' (Admin, Admin, Admin) and 'Caregiver' (Caregiver, Caregiver, Caregiver, Caregiver, Caregiver, Caregiver). On the right, there's a 'Username' field, 'Update Account' button, and 'Logout' link. Below the navigation is a title 'Mikey Test' and a subtitle 'Halloween Journal Entries'. A journal entry is displayed with the title 'I ate too much candy' and date '11/01/20'. The entry content is: 'I ate SOO much candy today! First I ate all of my Twix bars, because they are the best. Then I ate My twizzlers and skittles. I feel like im gona be sic. It was the best day ever!'. There are 'Edit' and 'Close' buttons below the entry. At the bottom are 'View' and 'Add' buttons, and a 'Messages' link.

The screenshot shows a behavior tracking interface for 'Donnie'. At the top, it says 'Donnie 1st'. Below that is a section titled 'My Behavior at School Today' with a star rating system for various activities:

Social Skills	★ ★ ★	Math Rotations	★ ★ ★
Community Circle	★ ★ ★	Recess	★ ★ ★
Large Group Reading	★ ★ ★	Specials	★ ★ ★
Reading Rotations	★ ★ ★	Read Aloud	★ ★ ★
Writing	★ ★ ★	Science/Social Studies	★ ★ ★
Lunch	★ ★ ★	End of Day	★ ★ ★

At the bottom left is a 'Back' button with a left arrow icon. At the bottom right are icons for a floppy disk, a plus sign, and an information symbol (i).

Mikey

3rd

My Behavior at School Today

Following Directions



Accepting Others that are
not Following Directions



Social Skills



Back



(i)

Leo

4th

My Behavior at School Today

Following Directions



Accepting Others that are
not Following Directions



Social Skills



Back



(i)

My Behavior at School Today

Social Skills ★ ★ ★

Math Rotations ★ ★ ★

Community Circle ★ ★ ★

Recess ★ ★ ★

Large Group Reading ★ ★ ★

Specials ★ ★ ★

Reading Rotations ★ ★ ★

Read Aloud ★ ★ ★

Writing ★ ★ ★

Science/Social Studies ★ ★ ★

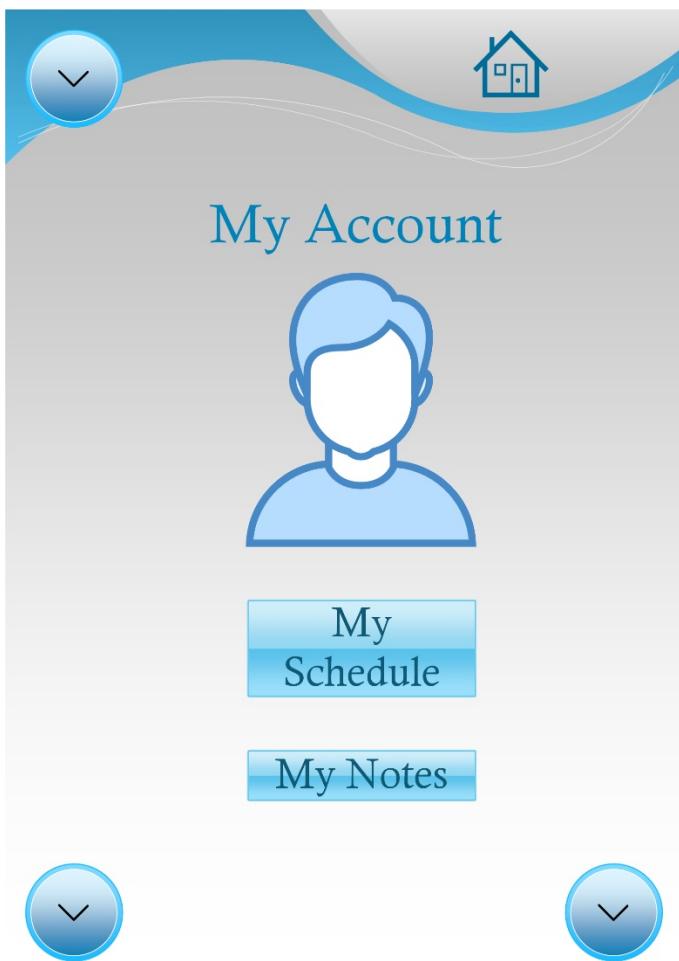
Lunch ★ ★ ★

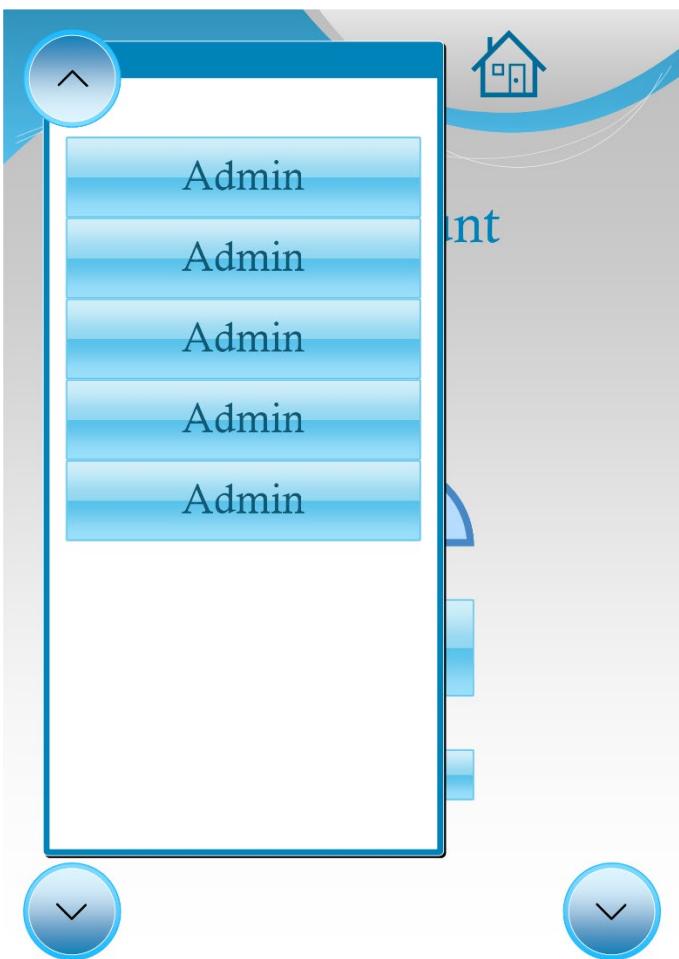
End of Day ★ ★ ★

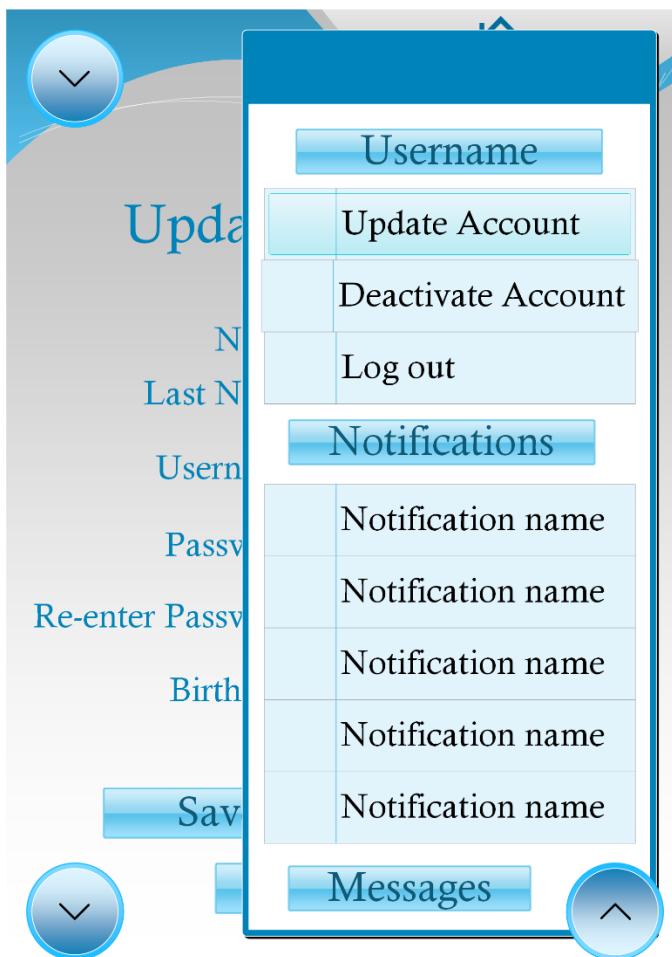
 Back

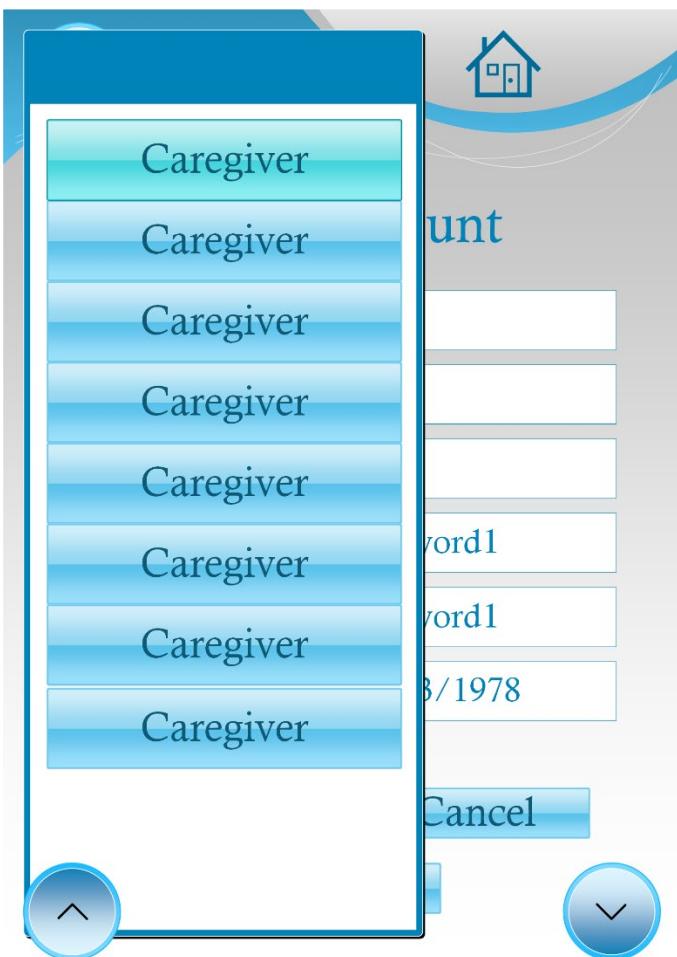


Mobile Portrait











Update Account

First Name:

Last Name:

Username:

Password:

Save Cancel

My Notes



Johnny Test

Date	A Note
Date	A Note
Date	A Note
Date	A Note

The image shows a digital communication interface. At the top left is a circular button with a downward arrow. At the top center is a house icon. Below the house is the name "Ms. Frizzle". In the center is a blue silhouette of a woman's head and shoulders. Below the silhouette is a blue button labeled "Send Message". Underneath the message button is a table with two rows. The first row has two columns: "Date" and "A Note". The second row also has two columns: "Date" and "A Note".

Date	A Note
Date	A Note



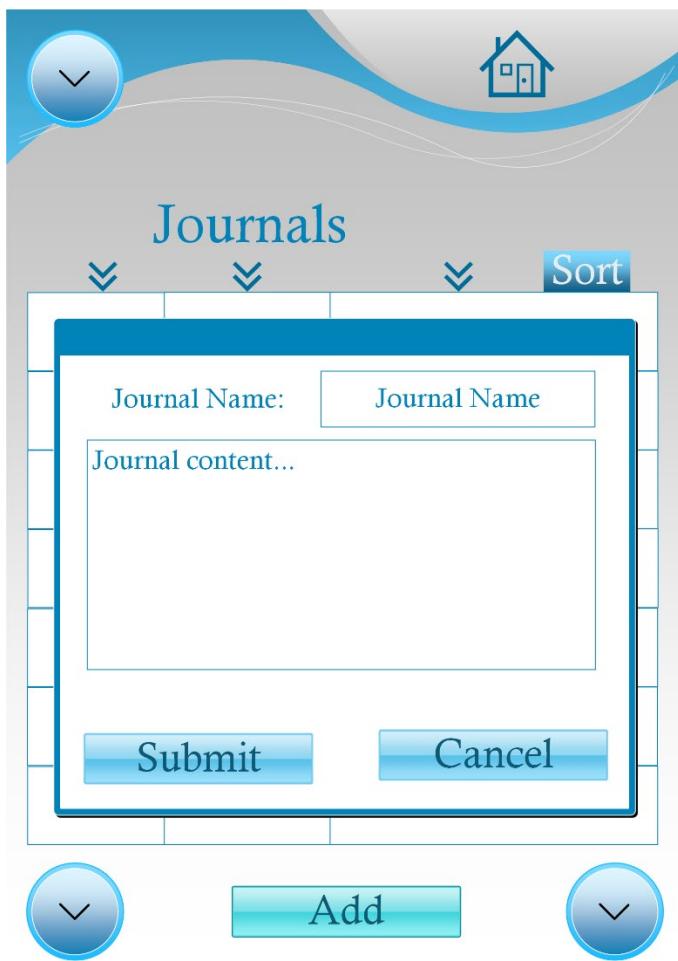
Date	Name	Desc.



Daily Tasks

Sort

Date	Name	Desc.





Messages		
▼	▼	▼ Sort
Date	Name	Desc.

Desktop

The application interface for 'Mikey Test' is shown. At the top, there is a navigation bar with 'File', 'Notes', 'Messages', and 'Help' buttons, along with a 'Username' dropdown and a gear icon. Below the navigation bar, there are two vertical lists of users: 'Admins' (Admin, Admin, Admin) and 'Caregivers' (Caregiver, Caregiver, Caregiver, Caregiver, Caregiver). The main area features a large blue profile picture of a person. Below the picture are several buttons: 'Goals', 'Routines', 'Journals', 'Progress', 'Rewards', and 'Tasks'. To the right of these buttons is a calendar for December 2020. Further down, there is a section titled 'Notifications:' with six entries, each labeled 'Notification'. A 'Messages' button is located at the bottom right.

The application interface for 'Mikey Test' is shown, specifically the 'Goals' section. The layout is identical to the desktop version above, with the same navigation bar, user lists, and main area. In the main area, the title 'Mikey Test' is followed by 'Goals'. Below the title are two buttons: 'Current' and 'Past'. Underneath these buttons is a list of goals. The first goal is 'I want to get a B+ on the Math test at the end of this month' with a progress bar from 11/01/20 to 11/01/20 and a status of 'Accomplished'. The second goal is 'I want to read a full chapter book on my own by the end of the school year' with a progress bar from 11/01/20 to 05/12/20 and a status of 'Accomplished'. To the right of the goals is a 'Notifications:' section with six entries labeled 'Notification'. A 'Messages' button is located at the bottom right.

File Notes Messages Help Username - X

Admins

- Admin
- Admin
- Admin

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

My Caregivers

	Mary Poppins
	Nanny
	Mrs. Doubtfire
	Nanny
	Ms. Frizzle
	Teacher
	Dr. Gregory House
	Doctor

December 20

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Notifications:

Notification

Messages

File Notes Messages Help Username - X

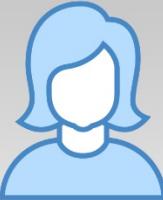
Admins

- Admin
- Admin
- Admin

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Ms. Frizzle



Send Message

December 20

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Notifications:

Notification

Messages

File Notes Messages Help Username  - X

Admins

- Admin
- Admin
- Admin

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Messages

Admins



Caregivers



You've been doing so great on your tasks! Keep up the good work buddy!

Thanks! I am trying to get to that next reward! I get a new game!

December 20						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

File Notes Messages Help Username  - X

Admins

- Admin
- Admin
- Admin

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Mikey Test Progress



[Report Cards](#)

[History](#)

Type	Date
Babysitter Report	11/01/20
Babysitter Report	10/31/20
School Behavior	10/30/20
School Behavior	10/29/20
School Behavior	10/28/20

December 20						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

Donnie

1st

My Behavior at School Today

Social Skills ★ ★ ★

Math Rotations ★ ★ ★

Community Circle ★ ★ ★

Recess ★ ★ ★

Large Group Reading ★ ★ ★

Specials ★ ★ ★

Reading Rotations ★ ★ ★

Read Aloud ★ ★ ★

Writing ★ ★ ★

Science/Social Studies ★ ★ ★

Lunch ★ ★ ★

End of Day ★ ★ ★



(i)

Mikey

3rd

My Behavior at School Today

Following Directions

1 2 3 4 5 6

Accepting Others that are
not Following Directions

1 2 3 4 5 6

Social Skills

1 2 3 4 5 6



(i)

Leo

4th

My Behavior at School Today

Following Directions



Accepting Others that are
not Following Directions



Social Skills



①

Donnie

1st

My Behavior at School Today

Social Skills



Math Rotations



Community Circle



Recess



Large Group Reading



Specials



Reading Rotations



Read Aloud



Writing



Science/Social Studies



Lunch



End of Day



①

Web

The image displays two side-by-side screenshots of a web application interface, both titled "Mikey Test".

Screenshot 1 (Top): This screenshot shows the main dashboard. On the left, there are two vertical lists: "Admins" (with three Admin entries) and "Caregivers" (with five Caregiver entries). The center features a user profile icon for "Mikey Test". Below the profile are six buttons arranged in two rows of three: Goals, Routines, Journals (top row); Progress, Rewards, Tasks (bottom row). To the right is a sidebar with a "Username" field, "Update Account" link, and "Logout" link. A "Notifications:" section lists six notifications, each with a "Notification" link. At the bottom right is a "Messages" button.

Screenshot 2 (Bottom): This screenshot shows the "Goals" section of the application. It has the same layout on the left with "Admins" and "Caregivers" lists. The center shows the user profile for "Mikey Test" with "Goals" sub-sections for "Current" and "Past". Below these are two goal cards:

- I want to get a B+ on the Math test at the end of this month
Progress: 11/01/20 - 11/01/20 (Accomplished)
- I want to read a full chapter book on my own by the end of the school year
Progress: 11/01/20 - 05/12/20 (Accomplished)

The right sidebar includes a "Username" field, "Update Account" link, "Logout" link, a "Notifications:" section with six notifications, and a "Messages" button.

Microsoft - Official Home + |

https://EnteryourWebsite.com/

The screenshot shows a web application interface. On the left, there is a sidebar with two sections: "Admins" containing three "Admin" entries and "Caregivers" containing five "Caregiver" entries. The main content area is titled "My Caregivers" and lists four caregivers with their roles: Mary Poppins (Nanny), Mrs. Doubtfire (Nanny), Ms. Frizzle (Teacher), and Dr. Gregory House (Doctor). To the right is a sidebar with a "Username" field, "Update Account" button, and "Logout" link. Below that is a "Notifications:" section listing six notifications, each with a "Notification" link. At the bottom is a "Messages" button.

Admins

Admin Admin Admin

Caregivers

Caregiver Caregiver Caregiver Caregiver Caregiver

My Caregivers

Mary Poppins Nanny

Mrs. Doubtfire Nanny

Ms. Frizzle Teacher

Dr. Gregory House Doctor

Username

Update Account

Logout

Notifications:

Notification

Notification

Notification

Notification

Notification

Notification

Messages

Microsoft - Official Home + |

https://EnteryourWebsite.com/

The screenshot shows a web application interface. On the left, there is a sidebar with two sections: "Admins" containing three "Admin" entries and "Caregivers" containing five "Caregiver" entries. The main content area is titled "Ms. Frizzle" and features a large blue profile picture of a woman with short blue hair. Below the picture is a "Send Message" button. To the right is a sidebar with a "Username" field, "Update Account" button, and "Logout" link. Below that is a "Notifications:" section listing six notifications, each with a "Notification" link. At the bottom is a "Messages" button.

Admins

Admin Admin Admin

Caregivers

Caregiver Caregiver Caregiver Caregiver Caregiver

Ms. Frizzle

Send Message

Username

Update Account

Logout

Notifications:

Notification

Notification

Notification

Notification

Notification

Notification

Messages

Microsoft - Official Home x + https://EnteryourWebsite.com/

Mikey Test Journals

Title	Date
Halloween Journal	10/2020
Bikes	09/2020
Mario	08/2020
I like Pokemon	07/2020
What should I go as?	06/2020

New Journal View Journal

Admin Admin Admin

Caregiver Caregiver Caregiver Caregiver Caregiver

Username

Update Account Logout

Notifications:

Notification Notification Notification Notification Notification Notification

Messages

Microsoft - Official Home x + https://EnteryourWebsite.com/

Mikey Test Halloween Journal Entries

Title	Date
I ate too much candy	11/01/20
My Halloween	10/31/20
I am so excited!!!	10/30/20
I tried on my new costume!	10/29/20
What should I go as?	10/28/20

View Add

Admin Admin Admin

Caregiver Caregiver Caregiver Caregiver Caregiver

Username

Update Account Logout

Notifications:

Notification Notification Notification Notification Notification

Messages

Microsoft - Official Home x + https://EnteryourWebsite.com/

Admins

Admin Admin Admin

Caregivers

Caregiver Caregiver Caregiver Caregiver Caregiver Caregiver

Mikey Test

Halloween Journal Entries

Title Date

I ate too much candy 11/01/20

View Journal Entry

Title: I ate too much candy

Journal:

I ate SOO much candy today! First I ate all of my Twix bars, because they are the best. Then I ate My twizzlers and skittles. I feel like im gona be sic. It was the best day ever!

Edit Close

View Add

Logout Notifications:

Notification Notification Notification Notification Notification Notification

Messages

This screenshot shows a web-based application interface. On the left, there are two vertical lists: 'Admins' (with three entries) and 'Caregivers' (with six entries). The main content area is titled 'Mikey Test' and 'Halloween Journal Entries'. A modal window is open, titled 'View Journal Entry', displaying a journal entry with the title 'I ate too much candy' and a detailed description about eating candy. Below the modal are buttons for 'Edit' and 'Close'. At the bottom of the main area are 'View' and 'Add' buttons. To the right, there's a sidebar with a user profile ('Username'), account management ('Update Account', 'Logout'), and a 'Notifications:' section listing six notifications. A 'Messages' button is also present.

Microsoft - Official Home x + https://EnteryourWebsite.com/

Admins

Admin Admin Admin

Caregivers

Caregiver Caregiver Caregiver Caregiver Caregiver Caregiver

Messages

Admins

Caregivers

You've been doing so great on your tasks! Keep up the good work buddy!

Thanks! I am trying to get to that next reward! I get a new game!

Send

Logout Notifications:

Notification Notification Notification Notification Notification Notification

Messages

This screenshot shows a web-based application interface. On the left, there are two vertical lists: 'Admins' (with three entries) and 'Caregivers' (with six entries). The main content area is titled 'Messages'. It shows two groups: 'Admins' (with three icons) and 'Caregivers' (with five icons, one of which is highlighted). Below the groups is a message box containing a message from an admin to a caregiver. The message says: 'You've been doing so great on your tasks! Keep up the good work buddy!'. The caregiver's response is: 'Thanks! I am trying to get to that next reward! I get a new game!'. At the bottom are 'Send' and 'Messages' buttons. To the right, there's a sidebar with a user profile ('Username'), account management ('Update Account', 'Logout'), and a 'Notifications:' section listing six notifications.

Microsoft - Official Home <https://EnteryourWebsite.com/>

Mikey Test

Progress



Report Cards
History

Type	Date
Babysitter Report	11/01/20
Babysitter Report	10/31/20
School Behavior	10/30/20
School Behavior	10/29/20
School Behavior	10/28/20

Username
Update Account
Logout

Notifications:

Notification

Messages

Admins
Admin
Admin
Admin

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver
Caregiver

Microsoft - Official Home <https://EnteryourWebsite.com/>

Donnie

1st

My Behavior at School Today

Social Skills	★ ★ ★	Math Rotations	★ ★ ★
Community Circle	★ ★ ★	Recess	★ ★ ★
Large Group Reading	★ ★ ★	Specials	★ ★ ★
Reading Rotations	★ ★ ★	Read Aloud	★ ★ ★
Writing	★ ★ ★	Science/Social Studies	★ ★ ★
Lunch	★ ★ ★	End of Day	★ ★ ★

Back 

Microsoft - Official Home x +

https://EnteryourWebsite.com/ Bookmarks ...

Mikey 3rd

My Behavior at School Today

Following Directions 1 2 3 4 5 6

Accepting Others that are not Following Directions 1 2 3 4 5 6

Social Skills 1 2 3 4 5 6

← Back Save +/- Info i

Microsoft - Official Home x +

https://EnteryourWebsite.com/ Bookmarks ...

Leo 4th

My Behavior at School Today

Following Directions 1 2 3

Accepting Others that are not Following Directions 1 2 3

Social Skills 1 2 3

← Back Save +/- Info i

Microsoft - Official Home x +

https://EnterYourWebsite.com/

Donnie 1st

My Behavior at School Today

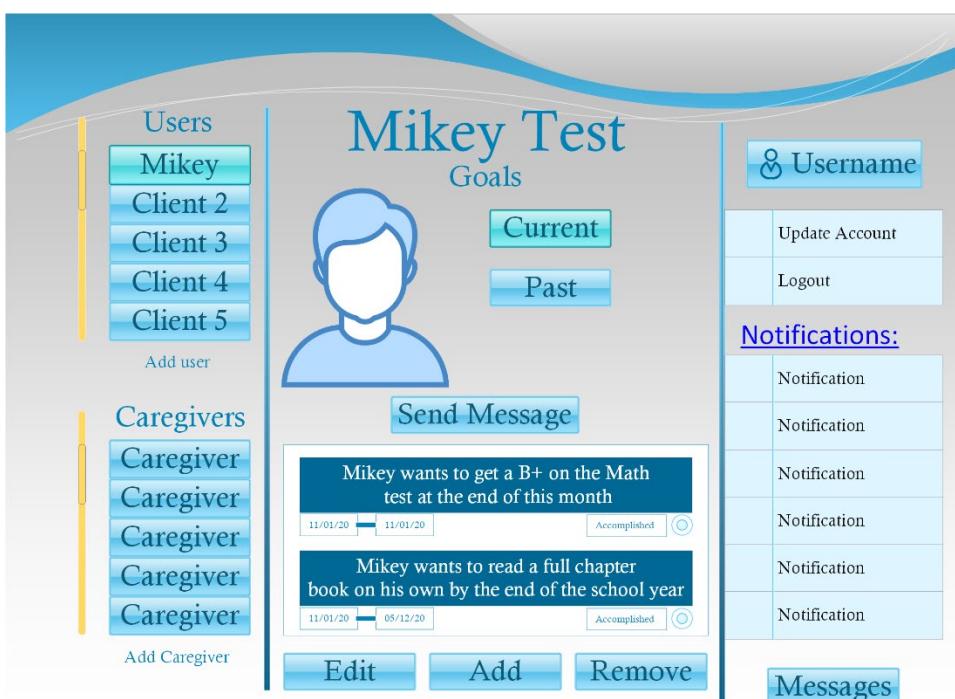
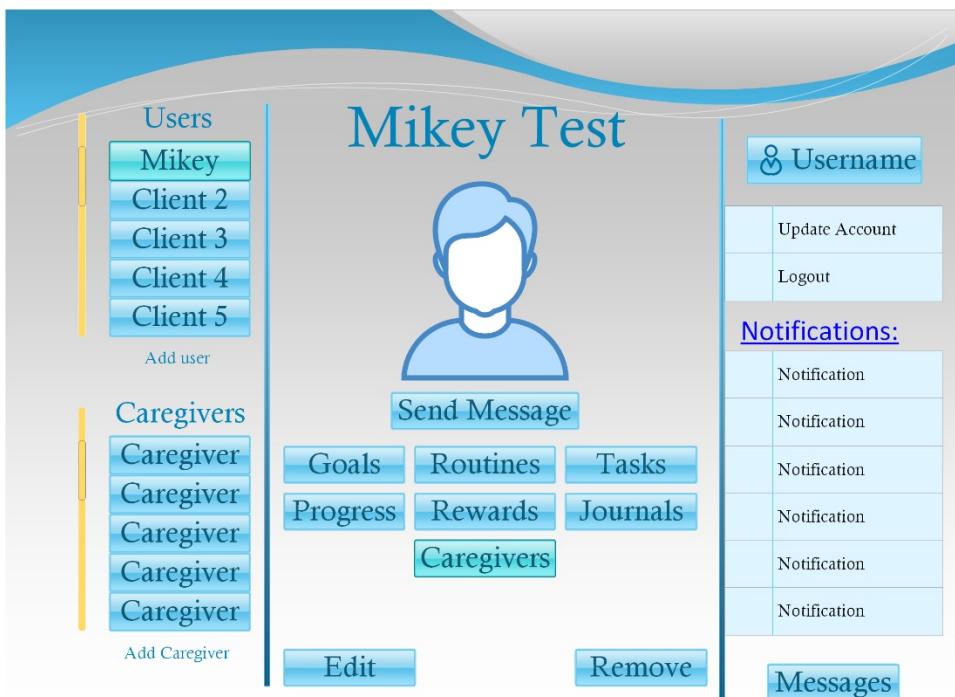
Social Skills	★ ★ ★	Math Rotations	★ ★ ★
Community Circle	★ ★ ★	Recess	★ ★ ★
Large Group Reading	★ ★ ★	Specials	★ ★ ★
Reading Rotations	★ ★ ★	Read Aloud	★ ★ ★
Writing	★ ★ ★	Science/Social Studies	★ ★ ★
Lunch	★ ★ ★	End of Day	★ ★ ★

 Back  

Administrative Interfaces

By Jory Wernette, Nick Loesel, and William Clark

Mobile Landscape



Mikey Test

Add Goals

Templates

Goal Name:

Description:

Related Incident:

Incident Time Frame:

Amount:

Notify: Admin Client Caregiver

Submit **Cancel**

Edit **Add** **Remove**

Users
Mikey
Client 2
Client 3
Client 4
Client 5
[Add user](#)

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver
Caregiver
[Add Caregiver](#)

Username

[Update Account](#)
[Logout](#)

Notifications:

- Notification

Messages

Mikey's Caregivers

Mary Poppins
Nanny

Mrs. Doubtfire
Nanny

Ms. Frizzle
Teacher

Dr. Gregory House
Doctor

Add

Users
Mikey
Client 2
Client 3
Client 4
Client 5
[Add user](#)

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver
Caregiver
[Add Caregiver](#)

Username

[Update Account](#)
[Logout](#)

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification
- Notification

Messages



Add User

Users	<input type="text" value="MMeyers"/>
Mikey	
Client 2	
Client 3	
Client 4	
Client 5	

Caregivers

Caregiver

Add user

Add Caregiver

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification
- Notification

Messages

Add Caregiver

Users	<input type="text" value="CTurk"/>
Mikey	
Client 2	
Client 3	
Client 4	
Client 5	

Caregivers

Caregiver

Add user

Add Caregiver

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification
- Notification

Messages

Users

- Mikey
- Client 2
- Client 3
- Client 4
- Client 5

Add user

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Add Caregiver

Messages

Users

Caregivers

Send

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification
- Notification

Username

Update Account

Logout

Messages

Users

- Mikey
- Client 2
- Client 3
- Client 4
- Client 5

Add user

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Add Caregiver

Mikey Test Progress

Type	Date
Babysitter Report	11/01/20
Babysitter Report	10/31/20
School Behavior	10/30/20
School Behavior	10/29/20
School Behavior	10/28/20

Send Message

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification

Username

Update Account

Logout

Messages

Donnie

1st

My Behavior at School Today

Social Skills ★ ★ ★

Math Rotations ★ ★ ★

Community Circle ★ ★ ★

Recess ★ ★ ★

Large Group Reading ★ ★ ★

Specials ★ ★ ★

Reading Rotations ★ ★ ★

Read Aloud ★ ★ ★

Writing ★ ★ ★

Science/Social Studies ★ ★ ★

Lunch ★ ★ ★

End of Day ★ ★ ★



Back



(i)

Mikey

3rd

My Behavior at School Today

Following Directions



Accepting Others that are
not Following Directions



Social Skills



Back



(i)

Leo

4th

My Behavior at School Today

Following Directions



Accepting Others that are
not Following Directions



Social Skills



Back



(i)

Donnie

1st

My Behavior at School Today

Social Skills ★ ★ ★

Math Rotations ★ ★ ★

Community Circle ★ ★ ★

Recess ★ ★ ★

Large Group Reading ★ ★ ★

Specials ★ ★ ★

Reading Rotations ★ ★ ★

Read Aloud ★ ★ ★

Writing ★ ★ ★

Science/Social Studies ★ ★ ★

Lunch ★ ★ ★

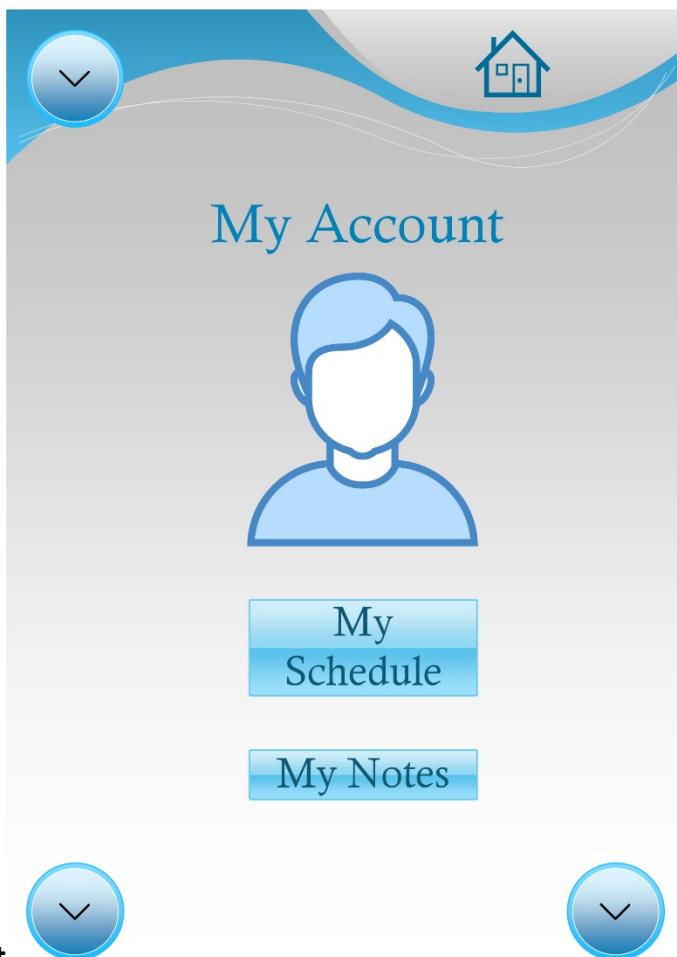
End of Day ★ ★ ★



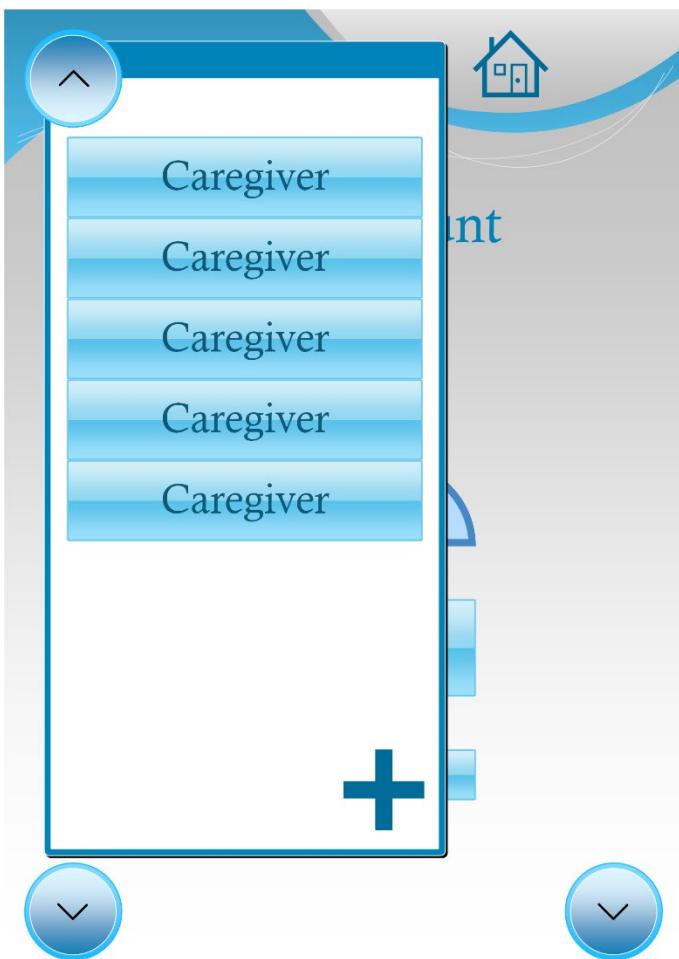
Back

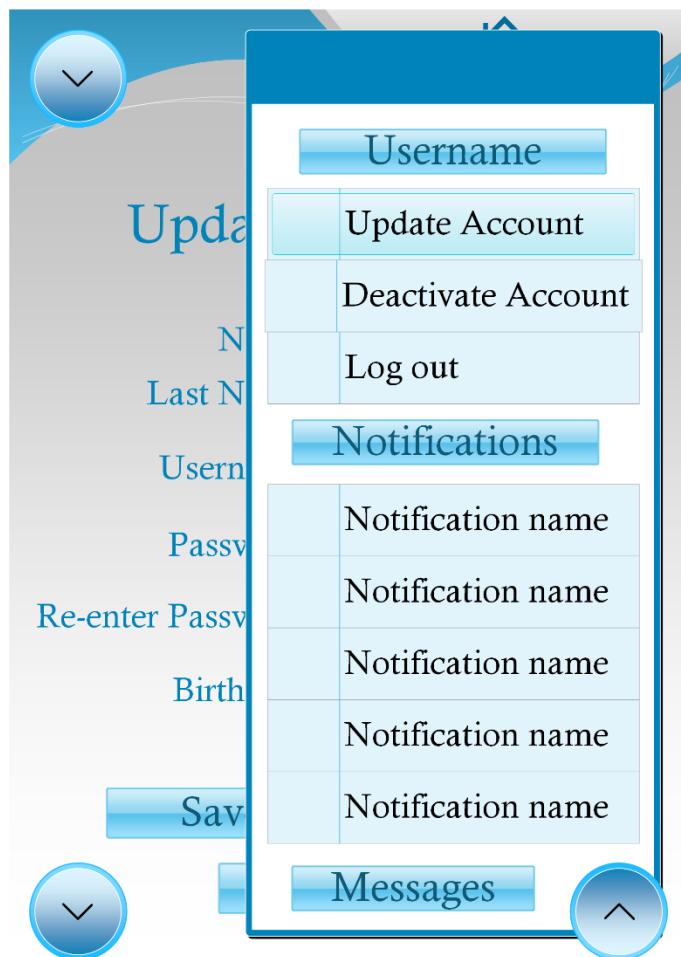


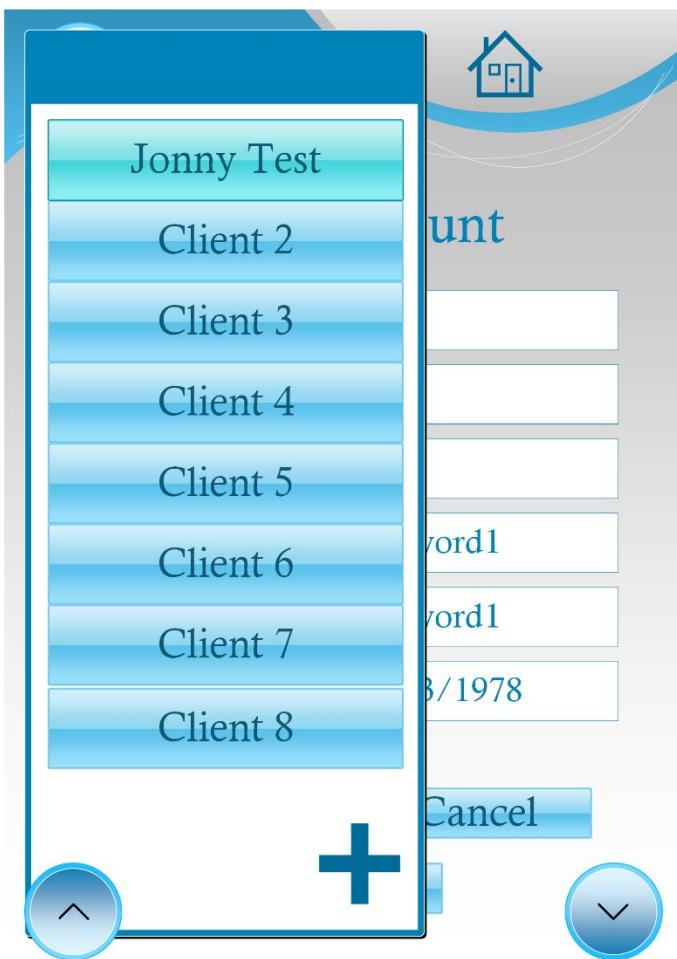
(i)

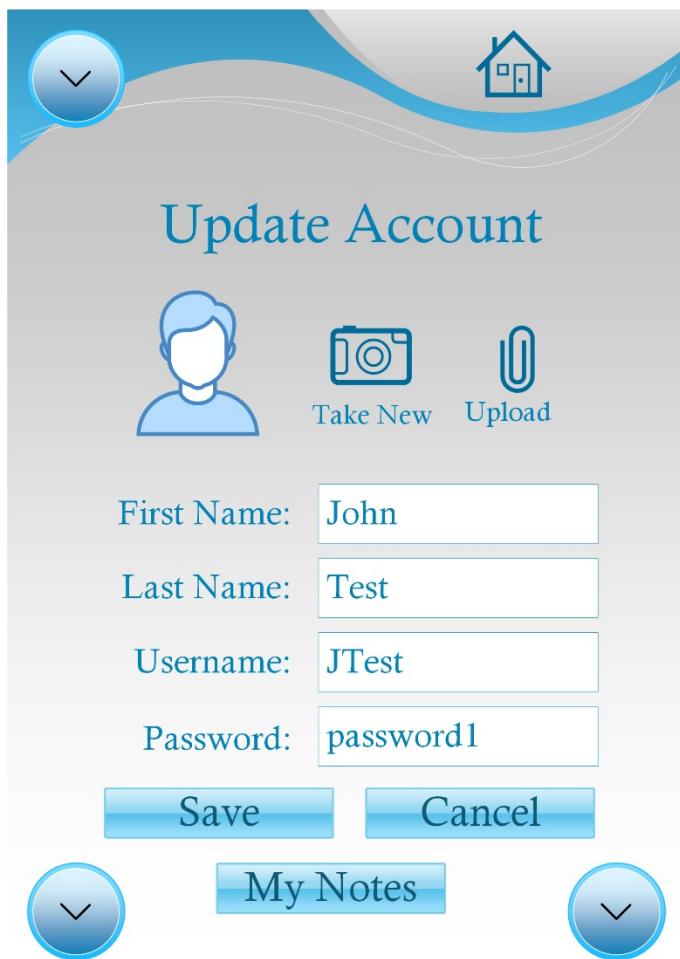


Mobile Portrait









A mobile application interface for updating account information. The top navigation bar includes a back arrow, a house icon, and a search icon. The main title is "Update Account". Below the title are three options: "Take New" (camera icon) and "Upload" (clip icon). The form fields are as follows:

First Name:	John
Last Name:	Test
Username:	JTest
Password:	password1

At the bottom are "Save" and "Cancel" buttons. A "My Notes" button is centered below the form.

First Name: John

Last Name: Test

Username: JTest

Password: password1

Save Cancel

My Notes

The screenshot shows the main profile screen of the Assistability app. At the top, there's a blue header bar with a circular arrow icon on the left and a house icon on the right. Below the header, the name "Johnny Test" is displayed in a large, bold, blue font. To the left of the name is a blue silhouette of a person's head and shoulders. To the right of the name are four blue rectangular buttons arranged in a 2x2 grid: "Goals" and "Progress" at the top, "Routines" and "Rewards" in the middle, and "Tasks" and "Journals" at the bottom. Below these buttons are three more blue rectangular buttons: "Edit", "Caregivers", and "Leave a note". Underneath these is a blue button labeled "Send Message". Below the message button is a table with two rows. The first row has two columns: "Date" and "A Note". The second row also has two columns: "Date" and "A Note". At the bottom of the screen are three blue circular icons with downward arrows: one on the left, one in the center containing a "Remove" button, and one on the right.

Date	A Note
Date	A Note

The image shows a mobile application interface with a light gray background. At the top center is a house icon. In the top left corner is a circular button with a downward arrow. In the top right corner is another circular button with a downward arrow. Below the house icon, the word "Goals:" is displayed in a large, bold, blue font. Underneath "Goals:" are four small blue downward arrows pointing downwards. To the right of these arrows is a blue rectangular button labeled "Sort". Below this section is a table with five rows. The table has three columns: "Date", "Name", and "Desc.". Each row contains placeholder text: "Date", "Name", and "Desc.". At the bottom of the table are two blue rectangular buttons: "Add" on the left and "Remove" on the right.

Date	Name	Desc.
Date	Name	Desc.

Add Remove

Create Goal

Goal Name:

Description:

Description

Related Incident:

Related Incident Information

Submit Cancel

Ms. Frizzle

Mikey

Client 2

Client 5

Assign Users

Edit Notes Permissions

Send Message

Date	A Note
Date	A Note

Remove



All Notifications:

Sort

Date	Name	Desc.



Messages

Sort

Date	Name	Desc.
Date	Name	Desc.

Desktop

File Notes Messages Help Username   - X

Users

- Mikey
- Client 2
- Client 3
- Client 4
- Client 5

[Add user](#)

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

[Add Caregiver](#)

Mikey Test



[Send Message](#)

Goals	Routines	Tasks
Progress	Rewards	Journals

[Caregivers](#)

[Edit](#) [Remove](#)

December 20						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

File Notes Messages Help Username   - X

Users

- Mikey
- Client 2
- Client 3
- Client 4
- Client 5

[Add user](#)

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

[Add Caregiver](#)

Mikey Test

Goals



[Current](#) [Past](#)

[Send Message](#)

Mikey wants to get a B+ on the Math test at the end of this month

11/01/20 — 11/01/20 Accomplished 

Mikey wants to read a full chapter book on his own by the end of the school year

11/01/20 — 05/12/20 Accomplished 

[Edit](#) [Add](#) [Remove](#)

December 20						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

File Notes Messages Help Username   

Mikey Test

Add Goals

Templates

Goal Name:	Goal Name		
Description:	Description		
Related Incident:	Related Incident		
Incident Amount:	Time Frame:		
Notify:	Admin <input checked="" type="checkbox"/>	Client <input type="checkbox"/>	Caregiver <input checked="" type="checkbox"/>

Submit **Cancel**

Edit **Add** **Remove**

December 20						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

File Notes Messages Help Username   

Mikey's Caregivers

 Mary Poppins
 Nanny
 Mrs. Doubtfire
 Nanny
 Ms. Frizzle
 Teacher
 Dr. Gregory House
 Doctor

Add

December 20						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

File Notes Messages Help Username

Ms. Frizzle

Assign Users

Mikey
Client 2
Client 3
Client 4
Client 5

Send Message

Notes

Edit Permissions

Remove

December 20

M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

Add user

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Add Caregiver

File Notes Messages Help Username

Ms. Frizzle

Edit Permissions

Mikey

Client 2

Client 5

Complete routines

Set goals

Adjust goals

Review history

Manage caregivers

Give rewards

Save **Cancel**

December 20

M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

Add user

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Add Caregiver

File Notes Messages Help Username

Add User

Find from Username

<input type="radio"/> Michael Meyers
<input type="radio"/> Mitchell Meyers
<input type="radio"/> Marie Meyers
<input type="radio"/> Mason Meyers
<input type="radio"/> Maverick Meyers

Add **Create Account for Someone** **Cancel**

December 20
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Notifications:

Notification

Users
Mikey
Client 2
Client 3
Client 4
Client 5
Add user

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver
Caregiver
Caregiver
Add Caregiver

File Notes Messages Help Username

Add Caregiver

Find from Username

<input type="radio"/> Christopher Turk
<input type="radio"/> Turk Turkleton
<input type="radio"/> Carrie Turkey
<input type="radio"/> Chase Turkeys
<input type="radio"/> Count Turk

Send Invitation **Create Account for Someone** **Cancel**

December 20
M T W T F S S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Notifications:

Notification

Users
Mikey
Client 2
Client 3
Client 4
Client 5
Add user

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver
Caregiver
Add Caregiver

File Notes Messages Help Username   

Users

- Mikey
- Client 2
- Client 3
- Client 4
- Client 5

[Add user](#)

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

[Add Caregiver](#)

Messages



You've been doing so great on your tasks!! Keep up the good work buddy!

Thanks! I am trying to get to that next reward! I get a new game!

[Send](#)

December 20						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

File Notes Messages Help Username   

Users

- Mikey
- Client 2
- Client 3
- Client 4
- Client 5

[Add user](#)

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

[Add Caregiver](#)

Mikey Test Progress



[Report Cards](#)

[History](#)

[Send Message](#)

Type: Babysitter Report Date: 11/01/20

Type: Babysitter Report Date: 10/31/20

Type: School Behavior Date: 10/30/20

Type: School Behavior Date: 10/29/20

Type: School Behavior Date: 10/28/20

December 20						
M	T	W	T	F	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Notifications:

Notification

Donnie

1st

My Behavior at School Today

Social Skills ★ ★ ★

Math Rotations ★ ★ ★

Community Circle ★ ★ ★

Recess ★ ★ ★

Large Group Reading ★ ★ ★

Specials ★ ★ ★

Reading Rotations ★ ★ ★

Read Aloud ★ ★ ★

Writing ★ ★ ★

Science/Social Studies ★ ★ ★

Lunch ★ ★ ★

End of Day ★ ★ ★



(i)

Mikey

3rd

My Behavior at School Today

Following Directions

1 2 3 4 5 6

Accepting Others that are
not Following Directions

1 2 3 4 5 6

Social Skills

1 2 3 4 5 6



(i)

Leo

4th

My Behavior at School Today

Following Directions



Accepting Others that are
not Following Directions



Social Skills



(i)

Donnie

1st

My Behavior at School Today

Social Skills



Math Rotations



Community Circle



Recess



Large Group Reading



Specials



Reading Rotations



Read Aloud



Writing



Science/Social Studies



Lunch



End of Day



(i)

Web

The image displays two screenshots of a web-based application interface, likely for a healthcare or educational platform. The left sidebar lists users and caregivers, while the main area shows a profile for 'Mikey Test' and various management tools.

Screenshot 1: Main Profile View

- Left Sidebar:**
 - Users:** Mikey, Client 2, Client 3, Client 4, Client 5
 - Caregivers:** Caregiver (repeated 5 times)
- Main Area:** Profile for **Mikey Test** (placeholder image). Buttons include:
 - Send Message
 - Goals, Routines, Tasks
 - Progress, Rewards, Journals
 - Caregivers
- Bottom Buttons:** Edit, Remove
- Right Sidebar:**
 - Username: Mikey
 - Update Account, Logout
 - Notifications:** (List of 6 notifications)
 - Messages

Screenshot 2: Goals View

- Left Sidebar:** Same as Screenshot 1.
- Main Area:** Profile for **Mikey Test** (placeholder image). Buttons include:
 - Send Message
 - Current, Past
- Content Area:** Two goal cards:
 - Mikey wants to get a B+ on the Math test at the end of this month (Timeline: 11/01/20 - 11/11/20, Status: Accomplished)
 - Mikey wants to read a full chapter book on his own by the end of the school year (Timeline: 11/01/20 - 05/12/20, Status: In Progress)
- Bottom Buttons:** Edit, Add, Remove
- Right Sidebar:** Same as Screenshot 1.

Microsoft - Official Home x + https://EnteryourWebsite.com/

Mikey Test

Add Goals

Templates

Goal Name:	Goal Name		
Description:	Description		
Related Incident:	Related Incident		
Incident Amount:	Time Frame:		
Notify:	Admin <input checked="" type="checkbox"/>	Client <input type="checkbox"/>	Caregiver <input checked="" type="checkbox"/>

Submit **Cancel**

Edit **Add** **Remove**

Users
Mikey
Client 2
Client 3
Client 4
Client 5
[Add user](#)

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver
[Add Caregiver](#)

Notifications:
Notification
Notification
Notification
Notification
Notification
Notification

Messages

Microsoft - Official Home x + https://EnteryourWebsite.com/

Mikey's Caregivers

 Mary Poppins Nanny
 Mrs. Doubtfire Nanny
 Ms. Frizzle Teacher
 Dr. Gregory House Doctor

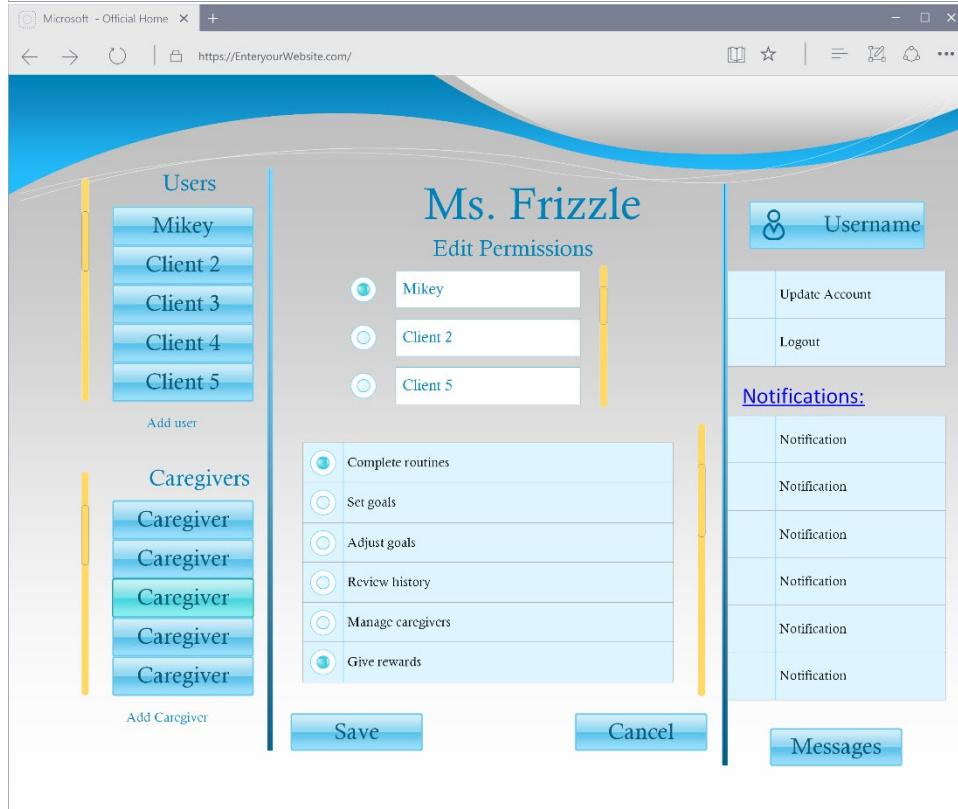
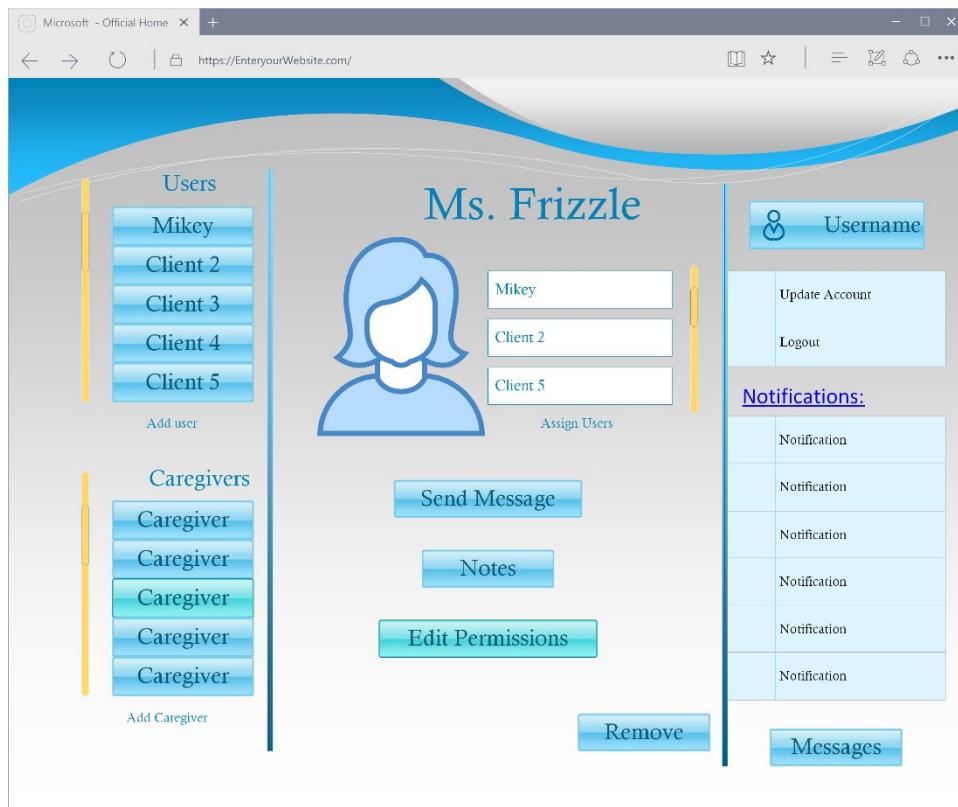
Add

Users
Mikey
Client 2
Client 3
Client 4
Client 5
[Add user](#)

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver
[Add Caregiver](#)

Notifications:
Notification
Notification
Notification
Notification
Notification
Notification

Messages



Microsoft - Official Home

https://EnteryourWebsite.com/

Add User

Find from Username: MMeyers

<input checked="" type="radio"/> Michael Meyers
<input type="radio"/> Mitchell Meyers
<input type="radio"/> Marie Meyers
<input type="radio"/> Mason Meyers
<input type="radio"/> Maverick Meyers

Add

Create Account for Someone

Cancel

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification
- Notification

Messages

Users
Mikey
Client 2
Client 3
Client 4
Client 5

Add user

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver

Add Caregiver

Microsoft - Official Home

https://EnteryourWebsite.com/

Add Caregiver

Find from Username: CTurk

<input checked="" type="radio"/> Christopher Turk
<input type="radio"/> Turk Turkleton
<input type="radio"/> Carrie Turkey
<input type="radio"/> Chase Turkeys
<input type="radio"/> Count Turk

Send Invitation

Create Account for Someone

Cancel

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification
- Notification

Messages

Users
Mikey
Client 2
Client 3
Client 4
Client 5

Add user

Caregivers
Caregiver
Caregiver
Caregiver
Caregiver

Add Caregiver

Microsoft - Official Home https://EnteryourWebsite.com/

Messages

Users

- Mikey
- Client 2
- Client 3
- Client 4
- Client 5

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Add user Add Caregiver

Username

Update Account Logout

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification
- Notification

Send Messages

You've been doing so great on your tasks! Keep up the good work buddy!

Thanks! I am trying to get to that next reward! I get a new game!

This screenshot shows a messaging interface. On the left, there are two vertical lists: 'Users' (Mikey, Client 2, Client 3, Client 4, Client 5) and 'Caregivers' (Caregiver, Caregiver, Caregiver, Caregiver, Caregiver). Below these are 'Add user' and 'Add Caregiver' buttons. The main area is titled 'Messages' and shows a list of users and caregivers with small profile icons. A message from a caregiver to 'Mikey' is highlighted with a yellow bar. The message content is: 'You've been doing so great on your tasks! Keep up the good work buddy!' and 'Thanks! I am trying to get to that next reward! I get a new game!'. On the right, there's a sidebar with 'Username', 'Update Account', and 'Logout' buttons. Below that is a 'Notifications:' section with six notifications. At the bottom are 'Send' and 'Messages' buttons.

Microsoft - Official Home https://EnteryourWebsite.com/

Mikey Test Progress

Users

- Mikey
- Client 2
- Client 3
- Client 4
- Client 5

Caregivers

- Caregiver
- Caregiver
- Caregiver
- Caregiver
- Caregiver

Add user Add Caregiver

Username

Update Account Logout

Notifications:

- Notification
- Notification
- Notification
- Notification
- Notification
- Notification

Report Cards History Send Message

Type Date

Type	Date
Babysitter Report	11/01/20
Babysitter Report	10/31/20
School Behavior	10/30/20
School Behavior	10/29/20
School Behavior	10/28/20

Messages

This screenshot shows a progress tracking interface for 'Mikey Test'. On the left, there are lists of 'Users' (Mikey, Client 2, Client 3, Client 4, Client 5) and 'Caregivers' (Caregiver, Caregiver, Caregiver, Caregiver, Caregiver). Below these are 'Add user' and 'Add Caregiver' buttons. The main area is titled 'Mikey Test Progress' and features a large blue profile icon of 'Mikey'. Below it are 'Report Cards' and 'History' buttons, and a 'Send Message' button. To the right of the profile are sections for 'Type' and 'Date' with a table of reports. The table has two columns: 'Type' (Babysitter Report, Babysitter Report, School Behavior, School Behavior, School Behavior) and 'Date' (11/01/20, 10/31/20, 10/30/20, 10/29/20, 10/28/20). On the right, there's a sidebar with 'Username', 'Update Account', and 'Logout' buttons. Below that is a 'Notifications:' section with six notifications. At the bottom is a 'Messages' button.

Microsoft - Official Home x + https://EnteryourWebsite.com/

Donnie 1st

My Behavior at School Today

Social Skills	★ ★ ★	Math Rotations	★ ★ ★
Community Circle	★ ★ ★	Recess	★ ★ ★
Large Group Reading	★ ★ ★	Specials	★ ★ ★
Reading Rotations	★ ★ ★	Read Aloud	★ ★ ★
Writing	★ ★ ★	Science/Social Studies	★ ★ ★
Lunch	★ ★ ★	End of Day	★ ★ ★

Back

Microsoft - Official Home x + https://EnteryourWebsite.com/

Mikey 3rd

My Behavior at School Today

Following Directions	1 2 3 4 5 6
Accepting Others that are not Following Directions	1 2 3 4 5 6
Social Skills	1 2 3 4 5 6

Back

Microsoft - Official Home  Back

https://EnteryourWebsite.com/

Leo 

My Behavior at School Today

Following Directions 1 2 3

Accepting Others that are not Following Directions 1 2 3

Social Skills 1 2 3

Microsoft - Official Home  Back

https://EnteryourWebsite.com/

Donnie 

My Behavior at School Today

Social Skills	  	Math Rotations	  
Community Circle	  	Recess	  
Large Group Reading	  	Specials	  
Reading Rotations	  	Read Aloud	  
Writing	  	Science/Social Studies	  
Lunch	  	End of Day	  

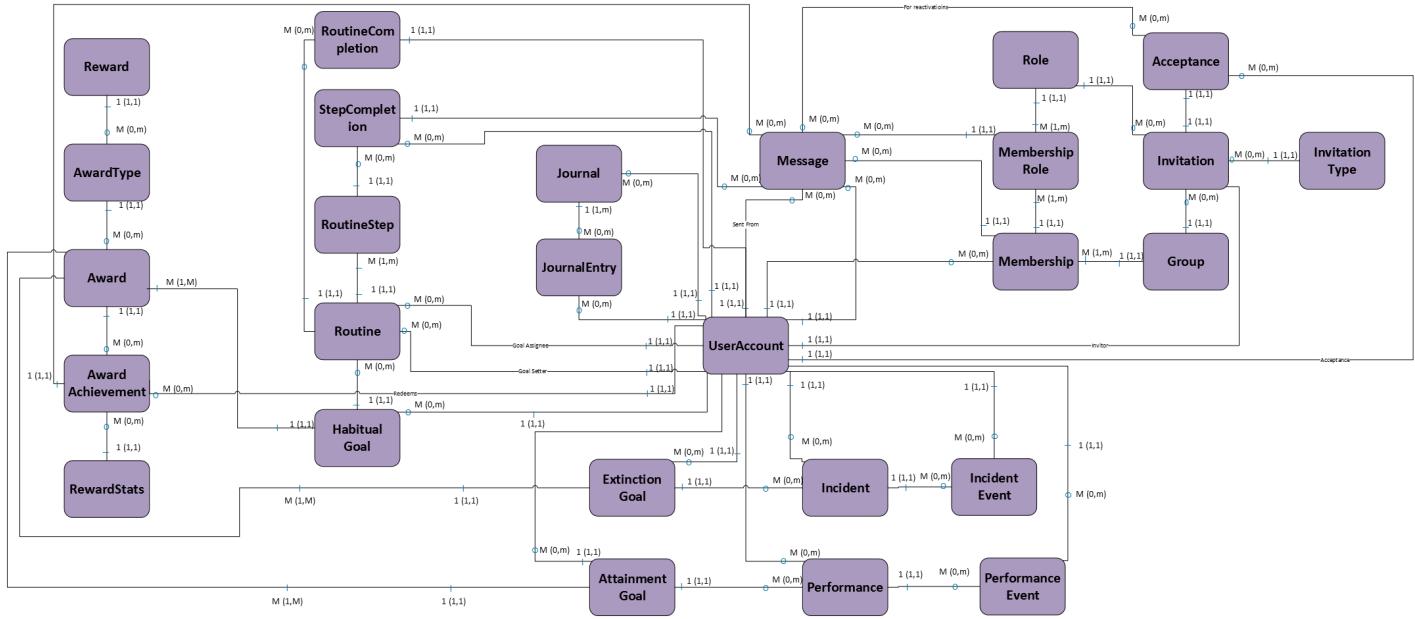
Data Layer

Implementation Assumptions

1. Application will be available for installation via public sources (app stores, open source)
2. Application will be deployable on Microsoft SQL Server 2017 and above
3. Will be easily portable to any ANSI-compliant SQL implementation.
4. Deployment scripts will be authored in TSQL.

E-R Diagram

By Nathaniel Webber contributions by: Becky Baenziger, William Clark, Mitchell Paul, Ryan Taylor, Whitney Vinson, Jory Wernette



Data Dictionary

By Whitney Vinson contributions by: Becky Baenziger, William Clark, Mitchell Paul, Ryan Taylor, Nathaniel Webber, Jory Wernette

FIELD	DATATYPE	LENGTH	DEFAULT	IDENTITY	START	INCREMENT	NULS	INDEX	UNIQUE	PK	FK	INTEGRITY	REFERENCES	COMMENTS
UserAccount														
UserID	INT			Yes	1	Auto	N	Y	Yes	Yes	No			The auto-generated User identification number. (V.2)
FirstName	NVARCHAR	50		No			N	N	No	No	No			The first name of the User.
LastName	NVARCHAR	50		No			N	N	No	No	No			The last name of the User.
Username	NVARCHAR	50		No			N	Y	Yes	No	No			The login name of the User.
PasswordHash	NVARCHAR	100		No			N	N	No	No	No			The password hash (SHA256).
Email	NVARCHAR	250		No			N	Y	Yes	No	No			The email address of the User.
Active	BIT			1	No		N	N	No	No	No			The active status of the User.
Role														
RoleID	INT			No	1	Auto	N	Y	Yes	Yes	No			The auto-generated Role identification number.
RoleName	NVARCHAR	50		No			N	Y	Yes	No	No			The name of the Role.
RoleDescription	NVARCHAR	100		No			N	N	No	No	No			The description of the Role.
RoleAccepted	DATETIME		DATETIME.NOW()	No			N	N	No	No	No			The date of Role acceptance.
Active	BIT			1	No		N	N	No	No	No			The active status of the Role.
Group														
GroupId	INT			Yes	1	Auto	N	Y	Yes	Yes	No			The auto-generated Group identification number.
MembershipRole														
UserID	INT			No			N	Y	Yes	No	Yes	Cascade Update	UserAccount.UserID	The unique identifier for the User.
RoleID	INT			No			N	Y	No	No	Yes	Cascade Update		The identifier for the Role.
Membership														
GroupID	INT			No			N	Y	No	Yes	Yes	Cascade Update	Group.GroupID	The identifier for the Group.
UserID	INT			No			N	Y	Yes	Yes	Yes	Cascade Update	UserAccount.UserID	The unique identifier for the User.
RoleID	INT			No			N	Y	No	Yes	Yes	Cascade Update	Role.RoleID	The identifier for the Role.
MembershipCreationDate	DATETIME		DATETIME.NOW()	No			N	Y	No	No	No			The date of Membership creation.
MembershipExpirationDate	DATETIME		No				N	Y	No	No	No			The date of Membership expiration.
Active	BIT			1	No		N	N	No	No	No			The active status of the Membership.
Invitation														
InvitationTypeName	NVARCHAR	50		No			N	N	No	No	Yes	Cascade Update	InvitationType.InvitationName	The name of the Invitation Type.
UserID_admin	INT			No			N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the admin who sent the Invitation.
InvitationRecipientID	INT			No			N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The temporary identifier for the recipient.
GroupID	INT			No			Y	Y	No	Yes	Yes	Cascade Update	Group.GroupID	The identifier for the assigned Group.
RoleID	INT			No			Y	Y	No	Yes	Yes	Cascade Update	Role.RoleID	The identifier for the assigned Role.
InvitationSentDate	DATETIME		DATETIME.NOW()	No			N	Y	No	Yes	No			The date the Invitation was sent.
InvitationExpirationDate	DATETIME		No				N	Y	No	No	No			The date of expiration for the Invitation.
InvitationType														
InvitationTypeName	NVARCHAR	50		No			N	Y	Yes	Yes	No			The name of the Invitation Type.
InvitationDescription	NVARCHAR	500		No			N	N	No	No	No			The description of the Invitation Type.
Acceptance														
InvitationSentDate	DATETIME		No				N	Y	No	Yes	No	Cascade Update	Invitation.InvitationSentDate	The date invitation to be accepted was sent
GroupID	INT		No				N	N	No	No	Yes	Cascade Update	Invitation.GroupID	The groupID of the invitation to be accepted
RoleID	INT		No				N	N	No	No	Yes	Cascade Update	Invitation.RoleID	The roleID of the invitation to be accepted
AcceptanceDate	DATETIME		No				Y	Y	No	No	No			The date the Invitation was accepted.
AcceptanceStatus	BIT		0	No			Y	N	No	No	No			The status of the Invitation.
Routine														
RoutineName	NVARCHAR	50		No			N	N	Yes	Yes	No			The name of the Routine.
RoutineDescription	NVARCHAR	150		No			N	N	No	No	No			The description of the Routine.
UserID_client	INT		No				N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the user assigned to the Routine.
UserID_admin	INT		No				N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the user who has set the Routine.
RoutineEntryDate	DATETIME		DATETIME.NOW()	No			N	Y	No	No	No			The date that the Routine was implemented.
RoutineEditDate	DATETIME		DATETIME.NOW()	No			Y	N	No	No	No			The date that the Routine was changed.
RoutineRemovalDate	DATETIME		DATETIME.NOW()	No			Y	N	No	No	No			The date that the Routine was archived.
Active	BIT		1	No			N	N	No	No	No			The active status of the Routine.
RoutineCompletion														
UserID_client	INT		No				N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The ID of the User (client) who has completed the Routine
RoutineName	NVARCHAR	50		No			N	N	Yes	No	Yes	Cascade Update	Routine.RoutineName	The name of the Routine.
RoutineDescription	NVARCHAR	150		No			N	N	No	No	No	Cascade Update	Routine.RoutineDescription	The description of the Routine.
RoutineCompletionDate	DATETIME		DATETIME.NOW()	No			N	Y	Yes	No	No			The date the Routine was completed.
RoutineStep														
RoutineStepID	INT		Yes	1	Auto		N	Y	Yes	Yes	No			The auto-generated Routine Step identification number.
RoutineName	NVARCHAR	50	No				N	N	Yes	No	Yes	Cascade Update	Routine.RoutineID	The identifier for the associated Routine.
RoutineStepName	NVARCHAR	50	No				N	Y	No	No	No			The name of the Routine Step.
RoutineStepDescription	NVARCHAR	150	No				N	N	No	No	No			The description of the Routine Step.
RoutineStepEntryDate	DATETIME		DATETIME.NOW()	No			N	N	No	No	No			The date of Routine Step creation.
RoutineStepEditDate	DATETIME		DATETIME.NOW()	No			N	Y	No	No	No			The date the Routine Step was changed.
RoutineStepRemovalDate	DATETIME		DATETIME.NOW()	No			N	Y	No	No	No			The date the Routine Step was archived.
RoutineStepOrderNumber	INT		No				N	N	Yes	No	No			The place of the Routine Step in the Routine order.
Active	BIT		1	No			N	N	No	No	No			The active status of the Routine Step.
RoutineStepCompletion														
UserID_client	INT		No				N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier of the User who completed a Routine Step.
RoutineStepID	INT		No				N	Y	No	No	Yes	Cascade Update	RoutineStep.RoutineStepID	The identifier for the Routine Step.
RoutineName	NVARCHAR	50	No				N	N	Yes	No	Yes	Cascade Update	Routine.RoutineID	The identifier for the associated Routine.
RoutineStepCompletionDate	DATETIME		DATETIME.NOW()	No			N	Y	Yes	No	No			The date the Routine Step was completed.
HabitualGoal														
UserID_client	INT		No				N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the User assigned to the Goal.
UserID_admin	INT		No				N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the User assigning a Goal.
HabGoalName	NVARCHAR	50	No				N	N	Yes	Yes	No			The name of the Goal.
HabGoalDescription	NVARCHAR	500	No				N	N	No	No	No			The description of the Goal.
HabGoalTargetDate	DATETIME		No				N	Y	No	No	No			The target date for achieving the Goal.
HabGoalEntryDate	DATETIME		DATETIME.NOW()	No			N	Y	No	No	No			The date that the Goal was implemented.
HabGoalEditDate	DATETIME		No				N	Y	No	No	No			The date that the Goal was changed.
HabGoalRemovalDate	DATETIME		No				N	Y	No	No	No			The date that the Goal was archived.
Active	BIT		0	No			N	N	No	No	No			The active status of the Goal.
AwardID	INT		No				N	Y	No	Yes	Yes	Cascade Update	Award.AwardID	The ID of the award given for achieving the Goal.
RoutineName	NVARCHAR	50	No				N	N	No	No	Yes	Cascade Update	Routine.RoutineName	The Name of the Routine involved

ExtinctionGoal												
ExtGoalName	NVARCHAR	50	No		N	N	Yes	Yes	No			The name of the Extinction Goal
ExtGoalDescription	NVARCHAR	500	No		N	N	No	No	No			The description of the Extinction Goal
IncidentName	NVARCHAR	50	No		N	N	No	No	Yes			The name of the Incident involved in the Goal
ExtGoalTargetDate	DATETIME	DATETIME.NOW()	No		N	N	No	No	No			The goal date for achieve the Extinction Goal
ExtGoalEntryDate	DATETIME		No		N	N	No	No	No			The date Extinction Goal was created
ExtGoalEditDate	DATETIME	No			Y	N	No	No	No			The date Extinction Goal was edited
ExtGoalRemovalDate	DATETIME	No			Y	N	No	No	No			The date Extinction Goal was removed
Active	BIT	0 No			N	N	No	No	No			Whether the goal is active or not
IncidentFrequency	INT		No		N	N	No	No	No			Frequency allowed for an Incident to achieve the Goal
UserIDClient	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The client using the Extinction Goal
UserIDCreator	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The user who created the Extinction Goal
AwardID	INT	No			N	N	No	No	Yes	Cascade Update	Award.AwardID	The AwardID for award when Extinction Goal is completed
Incident												
IncidentName	NVARCHAR	50	No		N	Y	No	Yes	No			The name of the Incident.
IncidentDescription	NVARCHAR	255	No		N	N	No	No	No			The description of the Incident.
DesiredConsequence	NVARCHAR	250	No		N	N	No	No	No			What should be done if the Incident occurs
IncidentEntryDate	DATETIME	DATETIME.NOW()	No		N	N	No	No	No			The date Incident was created
IncidentEditDate	DATETIME		No		Y	N	No	No	No			The date Incident was edited
IncidentRemovalDate	DATETIME	No			Y	N	No	No	No			The date Incident was removed
Active	BIT	0 No			N	N	No	No	No			Whether the Incident is active
UserIDClient	INT		No		N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the User.
UserIDCreator	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The user who created the Incident
IncidentEvent												
IncidentEventID	INT	Yes	1 Auto	N	N	Yes	Yes	No				The Id for the Incident event.
IncidentName	NVARCHAR		50	No	N	N	No	No	Yes	Cascade Update	Incident.IncidentName	The name of Incident involved
DateOfOccurrence	DATETIME	No			N	N	No	No	No			The date and time the Incident occurred
PersonsInvolved	NVARCHAR	250	No		N	N	No	No	No			A list of the persons involved
EventDescription	NVARCHAR	500	No		N	N	No	No	No			A description of the what happened in specific event
EventConsequence	NVARCHAR	250	No		N	N	No	No	No			What happened as a result of event
EventEditDate	DATETIME	No			Y	N	No	No	No			The date Incident was edit if edited
UserIDClient	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The user Incident was about
UserIDReporter	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The user that reported the incident
PerformanceEvent												
PerformanceEventID	INT	Yes	1 Auto	N	N	Yes	Yes	No				The Id for the preformance event.
PerformanceName	NVARCHAR		50	No	N	N	No	No	Yes	Cascade Update	Performance.PerformanceID	The name of the preformance.
DateOfOccurrence	DATETIME	No			N	N	No	No	No			The date the Prefomance happend.
PersonsInvolved	NVARCHAR	250	No		N	N	No	No	No			The list of people involved in the performance.
EventDescription	NVARCHAR	500	No		N	N	No	No	No			The description of the preformance event.
EventResult	NVARCHAR	250	No		N	N	No	No	No			The end result of the preformance event.
EventEditDate	DATETIME	No			Y	N	No	No	No			The date the preformance event was edited.
UserIDClient	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The user the performance event was about
UserIDReporter	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The user that reported the performance
Performance												
PerformanceName	NVARCHAR	50	No		N	Y	Yes	Yes	No			The name for the performance.
PerformanceDescription	NVARCHAR	255	No		N	N	No	No	Yes			The description of the performance.
UserID_client	INT	No			N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the tracked User.
UserIDCreator	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The user that created the performance
PerformanceEntryDate	DATETIME	DATETIME.NOW()	No		N	N	No	No	No			The date the Performance happend.
PerformanceEditDate	DATETIME		No		Y	N	No	No	No			The date the Prefomance was edited.
PerformanceRemovalDate	DATETIME	No			Y	N	No	No	No			The date the Performance was removed.
Active	BIT	0 No			N	N	No	No	No			The active status of the Performance.
AttainmentGoal	NVARCHAR		No		N	Y	Yes	Yes	No			The name of the attained Goal.
AttGoalDescription	NVARCHAR	255	No		N	N	No	No	No			The description of the attained Goal.
PerformanceName	INT	No			N	Y	No	No	Yes			The identifier for the associated Performance.
Active	BIT	0 No			N	N	No	No	No			The active status of the Attainment Goal.
AttGoalEntryDate	DATETIME		No		N	N	No	No	No			The date the Attainment Goal was created.
AttGoalEditDate	DATETIME	No			Y	N	No	No	No			The date the Attainment Goal was edited.
AttGoalRemovalDate	DATETIME	No			Y	N	No	No	No			The date the Attainment Goal was removed.
UserIDClient	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The ID of the User the Attainment Goal was created for.
UserIDCreator	INT	No			N	N	No	No	Yes	Cascade Update	UserAccount.UserID	The ID of the User that created the Attainment Goal.
AttGoalTargetDate	DATETIME	No			N	N	No	No	No			The date the Attainment Goal should be compleated
AwardID	INT	No			N	N	No	No	No	Cascade Update	Award.AwardID	The ID of the Award
Award												
AwardID	INT	Yes	1 Auto	N	Y	Yes	Yes	No				The unique identifier for the Award.
AwardName	NVARCHAR		50	No	N	N	No	No	No			The name of the Award.
AwardDescription	NVARCHAR	255	No		N	N	No	No	No			The description of the Award.
GoalID	INT	No			N	N	No	No	Yes	Cascade Update	Goal.GoalID	The identifier for the associated Goal.
GoalTypeID	INT	No			N	N	No	No	Yes	Cascade Update	GoalType.TypeID	The identifier for the associated Goal Type.
AwardType												
AwardTypeBronze	NVARCHAR	50	No		N	N	No	No	No			The Bronze award type.
AwardTypeSilver	NVARCHAR	50	No		Y	N	No	No	No			The Silver award type.
AwardTypeGold	NVARCHAR	50	No		N	N	No	No	No			The Gold award type.
RewardName	NVARCHAR	50	No		N	Y	No	No	Yes	Cascade Update	Reward.RewardName	The name for the Reward.
AwardAchievement												
AwardAchievementDate	DATETIME	DATETIME.NOW()	No		N	Y	No	No	No			The date the Award was achieved.
AwardID	INT	No			N	Y	No	No	Yes	Cascade Update	Award.AwardID	The identifier for the Award.
RewardID	INT	No			N	Y	No	No	Yes	Cascade Update	RewardStatus.RewardID	The identifier for the Reward.
Reward												
RewardID	INT	YES	1 Auto	N	Y	Yes	Yes	No				The unique identifier for the Reward earned.
RewardName	NVARCHAR	50	No		N	Y	Yes	Yes	No			The name for the Reward.
RewardDescription	NVARCHAR	255	No		N	N	No	No	No			The description for the Reward.
UserID_client	INT	No			N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the User earning the Reward.
UserID_admin	INT	No			N	Y	No	No	Yes	Cascade Update	UserAccount.UserID	The identifier for the User issuing the Reward.

RewardStatus															
RewardName	NVARCHAR	No			N	Y	No	No	Yes	Cascade Update	Reward.RewardName	The name for the Reward.			
RewardDateEarned	DATETIME	DATETIME.NOW()	No		N	N	No	No	No			The date the Award was earned.			
RewardRedeemedDate	DATETIME	No			Y	N	No	No	No			The date the Reward was used.			
RewardsRedeemed	BIT	0 No			N	N	No	No	No			The status of the Reward.			
Journal															
JournalName	NVARCHAR	50	No		N	Y	Yes	Yes	No			The auto-generated user identification number.			
JournalDescription	NVARCHAR	255	No		Y	N	No	No	No			The description of the Journal.			
UserID_client	INT		No		N	Y	No	Yes	Yes	Cascade Update	UserAccount.UserID	The identifier of the user writing an entry.			
JournalEntry															
UserID_client	INT		No		N	Y	Yes	Yes	Yes	Cascade Update	UserAccount.UserID	The identifier for the user writing an entry.			
UserID_ClientJournal	INT		No		N	N	No	No	Yes	Cascade Update	Journal.UserID_Client	The identifier for client of journal the journal entry is for			
JournalID	NVARCHAR	500	No		N	Y	No	No	Yes	Cascade Update	Journal.JournalName	The identifier for the associated Journal.			
JournalEntryBody	NVARCHAR	500	No		N	N	No	No	No			The body of the Journal Entry.			
JournalEntryDate	DATETIME	DATETIME.NOW()	No		N	Y	Yes	Yes	No			The date of the Journal Entry.			
JournalEditDate	DATETIME		No		Y	N	No	No	No			The date the Journal was edited.			
Message															
MessageID	INT		Yes	10000	Auto	N	Y	Yes	Yes	No		The unique identifier for the Message.			
UserID_sender	INT		No		N	N	No	Yes	Yes	Cascade Update	UserAccount.UserID	The identifier for the sending User.			
UserID_recipient	INT		No		N	N	No	Yes	Yes	Cascade Update	UserAccount.UserID	The identifier for the receiving User.			
MessageContent	NVARCHAR	2048	No		N	N	No	No	No			The content of the Message.			
MessageEntryDate	DATETIME	DATETIME.NOW()	No		N	N	No	No	No			The date of Message creation.			
MessageEditDate	DATETIME	DATETIME.NOW()	No		Y	N	No	No	No			The date the Message was edited.			
MessageArchiveDate	DATETIME	No			Y	N	No	No	No			The date the Message was archived.			
MessageReadStatus	BIT	0 No			Y	N	No	No	No			The read status of the sent Message.			

Data Operations – Stored Procedure Definitions

By Becky Baenziger contributions by: William Clark, Mitchell Paul, Ryan Taylor, Whitney Vinson, Nathaniel Webber, Jory Wernette

User

sp_create_user

```
(@UserID int, @FirstName nvarchar(50), @LastName nvarchar(50), @Username nvarchar(50),
@PasswordHash nvarchar(100), @Email nvarchar(250), @Active bit default 1)

-- insert user into UserAccount table
```

sp_edit_user

```
(@user_id int, @old_first_name(50), @old_last_name nvarchar(50), @OldUsername
nvarchar(50), @OldPasswordHash nvarchar(100), @OldEmail nvarchar(250), @NewFirstName
nvarchar(50), @NewLastName nvarchar(50), @NewUsername nvarchar(50),
@NewPasswordHash nvarchar(100), @NewEmail nvarchar(250))

-- returns int of rows affected
```

sp_deactivate_user

```
(@UserID int, @Active bit)

-- deactivate user in UserAccout table
```

sp_select_user_by_username

```
(@Username nvarchar(50))

-- select user by username
```

sp_select_user_by_lastname

```
(@LastName nvarchar (50))

-- select user by name
```

sp_select_user_by_email

```
(@Email nvarchar(250))

-- select user by email
```

Password

```
sp_update_password_hash  
(@oldPasswordHash nvarchar(100), @newPasswordHash nvarchar(100), @Email int)  
-- update password in UserAccount table
```

Login

```
sp_verify_username_and_password  
(@Username nvarchar(50), @PasswordHash nvarchar(100))  
-- returns one user with matching username and password_hash
```

Role

```
sp_create_role  
(@RoleName nvarchar(50), @RoleDescription nvarchar(100), @RoleAccepted datetime,  
@Active bit)  
-- insert role into Role table
```

```
sp_edit_role
```

```
(@RoleName nvarchar(50), @RoleDescription nvarchar(100), @RoleAccepted datetime,  
@Active bit)  
-- update role Role table
```

```
sp_deactivate_role
```

```
(@RoleID int, @Active bit)  
-- deactivate role in Role table
```

```
sp_select_all_roles
```

```
(@RoleID int)  
-- select all roles in Role table
```

```
sp_select_roles_by_active
```

```
(@Active bit)  
-- selects all active roles
```

Group

sp_create_group

(@GroupID int)

-- insert group into Group table

Membership

sp_create_membership

(@GroupID int, @UserID int, @RoleID int, @MembershipCreationDate datetime, @Active bit)

-- insert membership into Membership table

sp_edit_membership

(@GroupID int, @UserID int, @RoleID int, @MembershipCreationDate datetime,
@MembershipExpirationDate datetime, @Active bit)

-- insert membership into Membership table

sp_deactivate_membership

(@Active bit)

-- deactivate membership Membership table

sp_select_all_users_in_membership

(@UserID int)

-- select all users in membership table

sp_select_all_roles_in_membership

(@RoleID int)

-- select all roles in membership table

sp_select_all_groups_in_membership

(@GroupID int)

-- select all groups in Membership table

```
sp_select_all_roles_for_user_in_membership  
(@UserID int)  
-- select all roles for a user in Membership table
```

```
sp_select_membership_by_active  
(@Active bit)  
-- select all active memberships in Membership table
```

```
sp_select_all_memberships_with_expiration_date  
(@MembershipExpirationDate datetime)  
-- select all memberships that will expire
```

Membership Role

```
sp_select_all_roles_by(userID)  
(@UserID int)  
-- select all roles for userID
```

```
sp_select_all_users_by_roleID  
(@RoleID int)  
-- select all users for role
```

Journal

```
sp_create_journal  
(@JournalName nvarchar(50), @UserID int, @JournalDescription nvarchar(255))  
-- returns int of rows affected
```

```
sp_edit_journal  
(@oldJournalName nvarchar(50), @oldJournalDescription nvarchar(255), @newJournalName  
nvarchar(50), @newJournalDescription nvarchar(255) @UserID int)  
-- returns int of rows affected
```

```
sp_deactivate_journal
```

(@JournalName nvarchar(50), @Active bit)

-- removes active of journal in Journal table

sp_select_all_journals

(@UserID int)

-- returns all journals with entries based on the UserID from Journal table

sp_select_journal_by_name

(@journal_name nvarchar(50))

-- returns the journal with entries based on the journal name and UserID

Journal Entry

sp_create_journal_entry

(@JournalName nvarchar(50), @UserID int, @JournalEntryBody nvarchar(500))

-- returns int of rows affected

sp_edit_journal_entry

(@OldJournalName nvarchar(50), @OldJournalEditDate datetime, @OldJournalEntryBody
nvarchar(500), @NewJournalName nvarchar(50), @NewJournalEditDate datetime,
@NewJournalEntryBody nvarchar(500))

-- returns int of rows affected

sp_delete_journal_entry

(@JournalEntryName nvarchar(50))

-- delete journal entry from JournalEntry table

sp_select_journal_journal_entries

(@JournalName nvarchar(50), @UserID int, @JournalEntryName nvarchar(50))

-- returns the journal entries associated with journal_name

Message

sp_create_message

```
(@MessageID int, @UserIDSender int, @UserIDRecipient int, @MessageContent nvarchar(2048), @MessageEntryDate datetime, @MessageReadStatus)
```

```
-- inserts into message table
```

```
sp_view_conversation
```

```
(@UserIDSender int, @UserIDRecipient int, @MessageID)
```

```
-- select query from messageView view ordered by CreateMessageDate
```

```
sp_update_message_by_messageID
```

```
(@MessageID[int], @MessageContent[nvarchar](2048), @MessageEditDate datetime, @UserIDRecipient int, @MessageReadStatus)
```

```
-- update Message table returns @@ROWCOUNT
```

```
sp_deactivate_message
```

```
(@MessageID int, @Active bit)
```

```
-- deactivate message?
```

```
sp_delete_message
```

```
(@MessageID int)
```

```
-- delete message from Message table
```

Invitation

```
sp_create_invitation
```

```
(@InvitationTypeName nvarchar(50), @UserIDAdmin int, @InvitationRecipientID int, @GroupID int, @RoleID int, @InvitationSentDate datetime, @InvitationExpirationDate datetime)
```

```
-- insert acceptance into Acceptance table
```

```
sp_edit_invitation
```

```
(@InvitationTypeName nvarchar(50), @UserIDAdmin int, @InvitationRecipientID int, @GroupID int, @RoleID int, @InvitationSentDate datetime, @InvitationExpirationDate datetime)
```

```
-- update acceptance in Acceptance table
```

```
sp_deactivate_invitation  
(@InvitationExpirationDate datetime)  
-- deactivate invitation
```

```
sp_select_invitation_by_roleID  
(@RoleID int)  
-- select invitation by role
```

```
sp_select_invitation_by_invitation_recipientID  
(@InvitationRecipientID int)  
-- get invitation by recipient
```

```
sp_select_recipients_by_invitation_groupID  
(@GroupID int)  
-- get recipients of invitation by groupid
```

Invitation Type

```
sp_create_invitation_type  
(@InvitationTypeName nvarchar(50), @InvitationDescription nvarchar(500))  
-- insert invitationtype into InvitationType table
```

```
sp_edit_invitation_type  
(@InvitationTypeName nvarchar(50), @InvitationDescription nvarchar(500))  
-- update invitationtype in InvitationType table
```

Acceptance

```
sp_create_acceptance  
(@InvitationID int, @UserIDRecipient int, @Status bit, @AcceptanceDate DATETIME)  
-- insert an acceptance into acceptance table
```

```
sp_update_acceptance
(@InvitationID int, @UserIDRecipient int, @Status bit, @AcceptanceDate DATETIME)
-- update Acceptance table, return @@rowcount
```

```
sp_deactivate_acceptance
(@InvitationID int, @Status bit)
-- deactivate acceptance invite in Acceptance table
```

```
sp_select_acceptance_by_invitationID
(@InvitationID int)
-- get invitation from Acceptance table
```

Routine

```
sp_create_routine
(@RoutineID nvarchar(50), @RoutineDescription nvarchar(150), @UserIDClient int,
@UserIDAdmin int, @RoutineEntryDate datetime, @HabGoalID int @Active bit)
-- Insert routine into Routine table
```

```
sp_edit_routine
(@RoutineID nvarchar(50), @RoutineDescription nvarchar(150), @UserIDClient int,
@UserIDAdmin int, @RoutineEditDate datetime, @Active bit)
-- update routine in Routine table
```

```
sp_deactivate_routine
(@RoutineID nvarchar(50), @RoutineRemovalDate, @Active bit)
-- deactivate routine in Routine table
```

Routine Completion

```
sp_create_routine_completion
(@UserIDClient int, @RoutineID int, @RoutineDescription nvarchar(150),
@RoutineCompletionDate DATETIME)
-- insert into routineCompletion table
```

```
sp_select_all_routines_completed_by(userID  
(@UserIDClient int)  
-- select all routines completed from routine table
```

```
sp_select_single_routine_completion_by(userID  
(@UserID_Client[int], @RoutineID[int])  
-- select from routineCompletion table
```

Routine Step

```
sp_create_routine_step  
(@RoutineID nvarchar(50), @RoutineStepName nvarchar(50), @RoutineStepDescription  
nvarchar(150), @RoutineStepEntryDate datetime, @RoutineStepOrderNumber int, @Active bit)  
-- insert routine step into RoutineStep table
```

```
sp_edit_routine_step  
(@RoutineID nvarchar(50), @RoutineStepName nvarchar(50), @RoutineStepDescription  
nvarchar(150), @RoutineStepEditDate datetime, @RoutineStepOrderNumber int, @Active bit)  
-- update routine step into RoutineStep table
```

```
sp_deactivate_routine_step  
(@RoutineStepID int, @Active int)  
-- deactivate routine step in RoutineStep table
```

```
sp_select_all_routine_steps_in_routine  
(@RoutineID nvarchar(50))  
-- see all routine step in a routine
```

```
sp_select_all_routine_steps_in_routine_by_active  
(@RoutineID nvarchar(50), @Active bit)  
-- selects all routine steps still active for routine
```

Routine Step Completion

```
sp_create_routine_step_completion  
(@UserIDClient int, @RoutingStepID int, @CompletionDate DATETIME)  
-- insert into routineStepCompletion
```

```
sp_select_all_routine_step_completion_by(userID)
```

```
(@UserIDClient int)  
-- select all routineStepCompletion by userID
```

```
sp_select_all_routine_step_completed_by_routineID
```

```
(@RoutineID nvarchar(50))  
-- select all completed routine steps for routine
```

Habitual Goal

```
sp_create_habitual_goal  
(@UserIDClient int, @UserIDAdmin int, @HabGoalName nvarchar(50), @HabGoalDescription  
nvarchar(500), @HabGoalTargetDate datetime, @HabGoalEntryDate datetime,  
@HabGoalEditDate datetime, @HabGoalRemovalDate date, @Active bit, @AwardID int)  
-- insert habitualgoal into HabitualGoal table
```

```
sp_edit_habitual_goal
```

```
(@UserIDClient int, @UserIDAdmin int, @HabGoalName nvarchar(50), @HabGoalDescription  
nvarchar(500), @HabGoalTargetDate datetime, @HabGoalEntryDate datetime,  
@HabGoalEditDate datetime, @HabGoalRemovalDate date, @Active bit @AwardID int)  
-- update habitualgoal in HabitualGoal table
```

```
sp_deactivate_habitual_goal
```

```
(@Active bit)  
-- deactivate habitualgoal in HabitualGoal table
```

```
sp_select_habitual_goal_by_active
```

```
(@Active bit)
```

-- selects habitual goals by active status

sp_select_habitual_goals_by(userID_client

(@UserIDClient int)

-- select all habitual goals for client

sp_select_habitual_goals_by_target_date

(@HabGoalTargetDate datetime)

-- select all habitual goals with specific target date

sp_select_all_clients_by_goalname

(@HabGoalName nvarchar(50))

-- selects all clients with the same habitual goal

Extinction Goal

sp_create_extinction_goal

(@ExtGoalName nvarchar(50), @ExtGoalDescription nvarchar(500), @IncidentName
nvarchar(50), @ExtGoalTargetDate datetime, @ExtGoalEntryDate datetime, @ExtGoalEditDate
datetime, @Active bit, @IncidentFrequency int, @UserIDClient int, @UserIDCreator int,
@AwardID int)

-- insert extinction goal into Extinction table

sp_edit_extinction_goal

(@ExtGoalName nvarchar(50), @ExtGoalDescription nvarchar(500), @IncidentName
nvarchar(50), @ExtGoalTargetDate datetime, @ExtGoalEntryDate datetime, @ExtGoalEditDate
datetime, @Active bit, @IncidentFrequency int, @UserIDClient int, @UserIDCreator int,
@AwardID int)

-- update extinction goal in Extinction table

sp_deactivate_extinction_goal

(@Active bit)

-- deactivate extinction goal

```
sp_select_all_extinction_goal  
(@ExtinctionGoalName nvarchar(50))  
-- select all extinction goals
```

```
sp_select_extinction_goal_by_active  
(@Active bit)  
-- select all active goals
```

```
sp_select_extinction_by_client  
(@UserIDClient int)  
-- select all goal by client
```

```
sp_select_goal_by_target_date  
(@ExtGoalTargetDate datetime)  
-- select goals with target date
```

Incident

```
sp_create_incident  
(@IncidentName nvarchar(50), @IncidentDescription nvarchar(255), @DesiredConsequence  
nvarchar(250), @IncidentEntryDate datetime, @Active bit, @UserIDClient int, @UserIDCreator  
int)  
-- insert incidents into Incident table
```

```
sp_edit_incident  
(@IncidentName nvarchar(50), @IncidentDescription nvarchar(255), @DesiredConsequence  
nvarchar(250), @IncidentEntryDate datetime, @IncidentEntryDate datetime, @Active bit,  
@UserIDClient int, @UserIDCreator int)  
-- insert incidents into Incident table
```

```
sp_deactivate_incident
```

```
(@Active bit, @IncidentRemovalDate datetime)
```

```
-- deactivate incidents
```

```
sp_select_all_incidents_by_active
```

```
(@Active bit)
```

```
-- select all incidents that are active
```

```
sp_select_all_incidents_for_client
```

```
(@UserIDClient int)
```

```
-- select incident for a client
```

```
sp_select_all_incidents
```

```
(@IncidentName nvarchar(50))
```

```
-- select all incidents
```

```
sp_select_all_incident_events_for_extinction_goal
```

```
(@ExtGoalName nvarchar(50))
```

```
-- select all incidents for an extinction goal
```

```
sp_count_incidents_events_for_extinction_goal
```

```
(@ExtGoalName nvarchar(50))
```

```
-- get count of incidents for an extinction goal
```

Incident Event

```
sp_create_incident_event
```

```
(@IncidentName nvarchar(50), @DateOfOccurrence datetime, @PersonsInvolved nvarchar(250),
@EventDescription nvarchar(500), @EventConsequence nvarchar(250), @UserIDClient int,
@UserIDReporter int)
```

```
-- insert event into Event table
```

```
sp_edit_incident_event
```

```
(@IncidentName nvarchar(50), @DateOfOccurrence datetime, @PersonsInvolved nvarchar(250),
@EventDescription nvarchar(500), @EventConsequence nvarchar(250), @UserIDClient int,
@UserIDReporter int, @EventEditDat datetime)
```

```
-- update event in Event table
```

```
sp_select_all_events_by_client
```

```
(@UserIDClient int)
```

```
-- select all events by client
```

```
sp_select_all_events_by_reporter
```

```
(@UserIDReporter int)
```

```
-- select all events reported by user
```

```
sp_select_all_events_by_date
```

```
(@DateOfOccurrence datetime)
```

```
-- select all events by date
```

```
sp_select_all_events_by_incident_name
```

```
(@IncidentName nvarchar(50))
```

```
-- select all events by incident name
```

```
sp_select_all_events_for_incident
```

```
(@IncidentName nvarchar(50))
```

```
-- get all events for an incident name
```

```
sp_count_events_for_incident
```

```
(@IncidentName nvarchar(50))
```

```
-- get count of events for an incident
```

```
sp_date_range_of_events_for_incident  
(@IncidentName nvarchar(50), @StartDateOfOccurrence datetime, @EndDateOfOccurrence  
datetime)  
-- get events in date range
```

Attainment Goal

```
sp_create_attainment_goal  
(@AttGoalName nvarchar(50), @AttGoalDescription nvarchar(255), @PerformanceName  
nvarchar(50), @Active bit, @AttGoalEntryDate datetime, @UserIDClient int, @UserIDCreator  
int, @AttGoalTargetDate datetime, @AwardID int)  
-- insert attainment goal into AttainmentGoal table
```

```
sp_edit_attainment_goal  
(@AttGoalName nvarchar(50), @AttGoalDescription nvarchar(255), @PerformanceName  
nvarchar(50), @Active bit, @AttGoalEditDate datetime, @UserIDClient int, @UserIDCreator int,  
@AttGoalTargetDate datetime, @AwardID int)  
-- update attainment goal in AttainmentGoal table
```

```
sp_deactivate_attainment_goal  
(@Active bit, @AttGoalRemovalDate datetime)  
-- deactivate attainment goal
```

```
sp_select_performance_name_by_attainment_goal  
(@AttGoalName nvarchar(50))  
-- get all performances by attainment goal
```

Performance

```
sp_create_performance  
(@PerformanceName nvarchar(50), @PerformanceDescription nvarchar(255), @UserIDClient  
int, @UserIDCreator int, @PerformanceEntryDate datetime, @Active bit,  
@PerformanceEventTarget int)  
-- insert performance into Performance table
```

```
sp_edit_performance
```

```
(@PerformanceName nvarchar(50), @PerformanceDescription nvarchar(255), @UserIDClient  
int, @UserIDCreator int, @PerformanceEditDate datetime, @Active bit,  
@PerformanceEventTarget int)
```

```
-- update performance in Performance table
```

```
sp_deactivate_performance
```

```
(@Active bit, @PerformanceRemovalDate date)
```

```
-- deactivate performance
```

```
sp_select_all_performance_by_client
```

```
@UserIDClient int)
```

```
-- get all prformance by client
```

```
sp_select_all_performances_by_active
```

```
(@Active bit)
```

```
-- select all performances that are active
```

```
sp_select_all_performances
```

```
(@IncidentName nvarchar(50))
```

```
-- select all incidents
```

```
sp_select_all_performance_events_for_attainment_goal
```

```
(@ExtGoalName nvarchar(50))
```

```
-- select all incidents for an extintion goal
```

```
sp_count_performance_events_for_attainment_goal
```

```
(@ExtGoalName nvarchar(50))
```

```
-- get count of incidents for an extinction goal
```

Performance Event

```
sp_create_performance_event
```

```
(@IncidentName nvarchar(50), @DateOfOccurrence datetime, @PersonsInvolved nvarchar(250),
@EventDescription nvarchar(500), @EventConsequence nvarchar(250), @UserIDClient int,
@UserIDReporter int)
```

```
-- insert event into Event table
```

sp_edit_performance_event

```
(@IncidentName nvarchar(50), @DateOfOccurrence datetime, @PersonsInvolved nvarchar(250),
@EventDescription nvarchar(500), @EventConsequence nvarchar(250), @UserIDClient int,
@UserIDReporter int, @EventEditDat datetime)
```

```
-- update event in Event table
```

sp_select_all_performance_events_by_client

```
(@UserIDClient int)
```

```
-- select all performance events by client
```

sp_select_all_performance_events_by_reporter

```
(@UserIDReporter int)
```

```
-- select all performance events reported by user
```

sp_select_all_performance_events_by_date

```
(@DateOfOccurrence datetime)
```

```
-- select all events by date
```

sp_select_all_performance_events_by_incident_name

```
(@PerformanceName nvarchar(50))
```

```
-- select all events by incident name
```

sp_count_performance_events_for_incident

```
(@PerformanceName nvarchar(50))
```

```
-- get count of events for an incident
```

```
sp_date_range_of_performance_events_for_performance  
(@IncidentName nvarchar(50), @StartDateOfOccurrence datetime, @EndDateOfOccurrence  
datetime)  
-- get events in date range
```

Award

```
sp_create_award  
(@AwardName nvarchar(50), @AwardDescription nvarchar(500),  
@GoalID int, @GoalTypeID int)  
-- insert into Award table, return @@ROWCOUNT
```

```
sp_edit_award
```

```
(@AwardName nvarchar(50), @AwardDescription nvarchar(500), @GoalID int, @GoalTypeID  
int)  
-- update into Award table, return @@ROWCOUNT
```

```
sp_safely_deactivate_award
```

```
(@AwardID int)  
-- delete award from award, return @@rowcount
```

```
sp_select_award_by_awardID
```

```
(@AwardID int, @AwardName nvarchar(50))  
-- select award by awardID
```

```
sp_select_award_by_goalID
```

```
(@GoalID int)  
-- select all awards for GoalID
```

```
sp_select_all_awards
```

```
(@AwardID int)
```

-- select all awards from award table, ordered by awardID ASC

Award Achievement

sp_create_award_achievement

(@AwardID int, @RewardID int, @AwardAchievementDate datetime)

-- insert award achievement into AwardAchievement table

sp_edit_award_achievement

(@AwardID int, @RewardID int)

-- update award achievement

sp_delete_award_achievement

(@AwardID int, @RewardID int)

-- delete award achievement

sp_select_awards_achieved_in_date_range

(@BeginAwardAchievementDate datetime, @EndAwardAchievementDate datetime)

-- get awards achieved over time period

sp_select_awards_by_rewardID

(@RewardID int)

-- get all awards that result in a certain reward

Reward

sp_create_reward

(@RewardName nvarchar(50), @RewardDescription nvarchar(255), @UserIDClient int, @UserIDAdmin int)

-- insert reward into Reward table

sp_edit_reward

(@RewardName nvarchar(50), @RewardDescription nvarchar(255), @UserIDClient int, @UserIDAdmin int)

-- update reward into Reward table

sp_deactivate_reward

(@RewardID int, @Active bit)

-- deactivate reward

sp_select_all_rewards

(@RewardId int)

-- select all rewards

sp_select_reward_by(userID_client

(@UserIDClient int)

-- select rewards for a client

Reward Status

sp_create_reward_status

(@RewardName nvarchar(50), @RewardDateEarned datetime, @RewardRedeemedDate null,
@RewardIsRedeemed bit)

-- insert reward status into RewardStatus table

sp_use_reward

(@RewardName nvarchar(50), @RewardRedeemedDate datetime, @RewardIsRedeemed bit)

-- update reward status in RewardStatus table when reward redeemed

sp_select_available_rewards

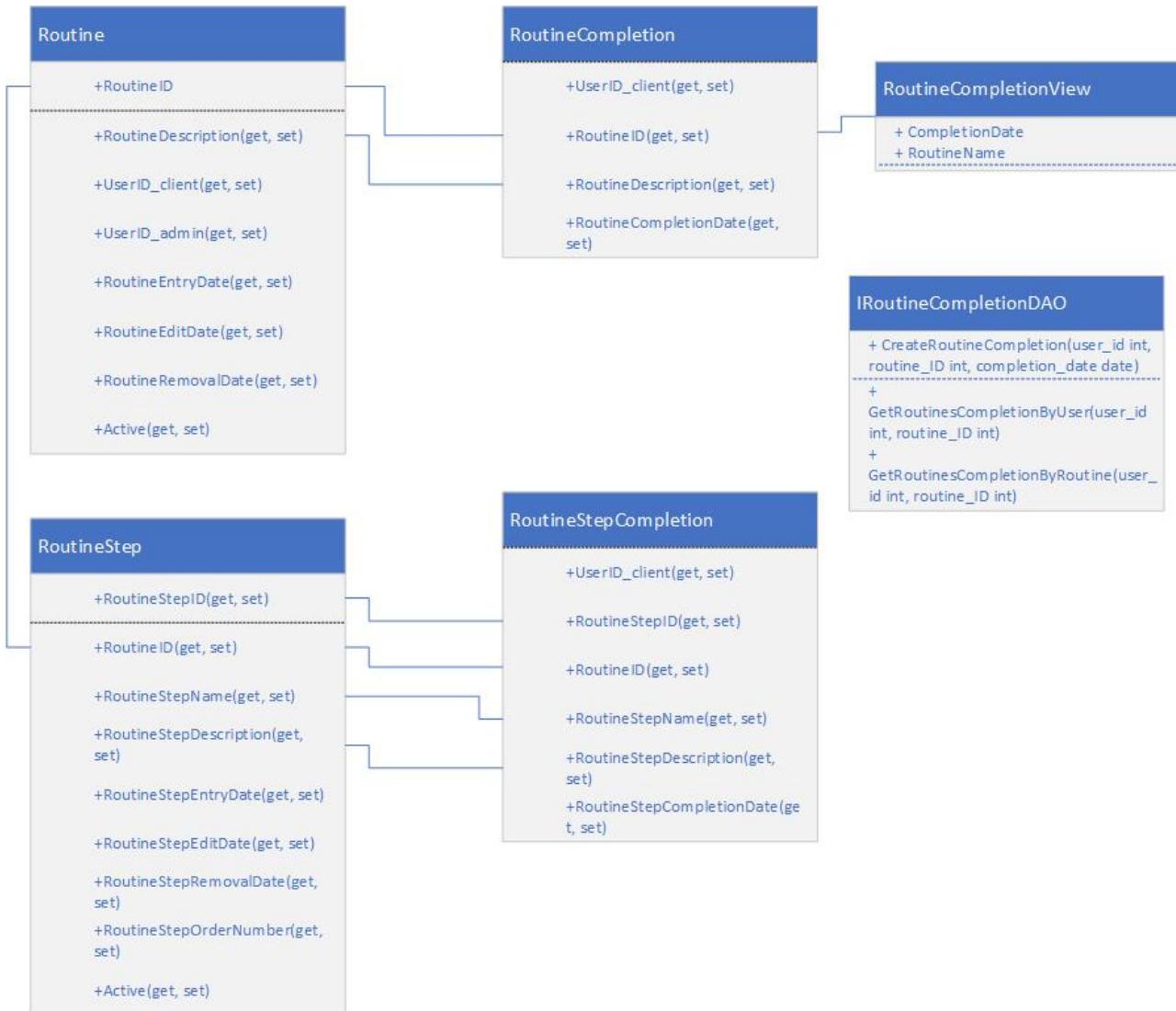
(@RewardName nvarchar(50), @RewardIsRedeemed bit)

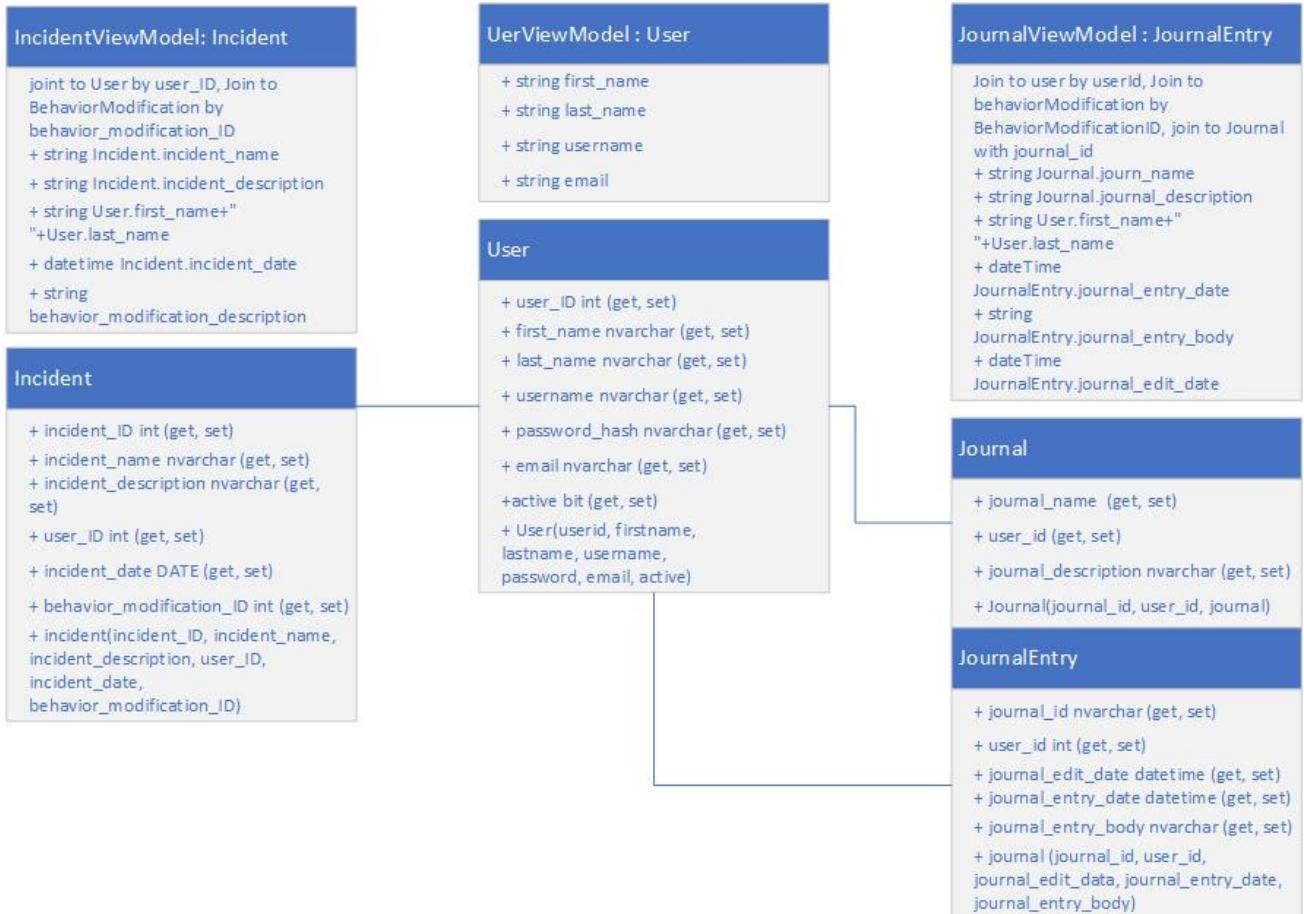
-- select all available rewards

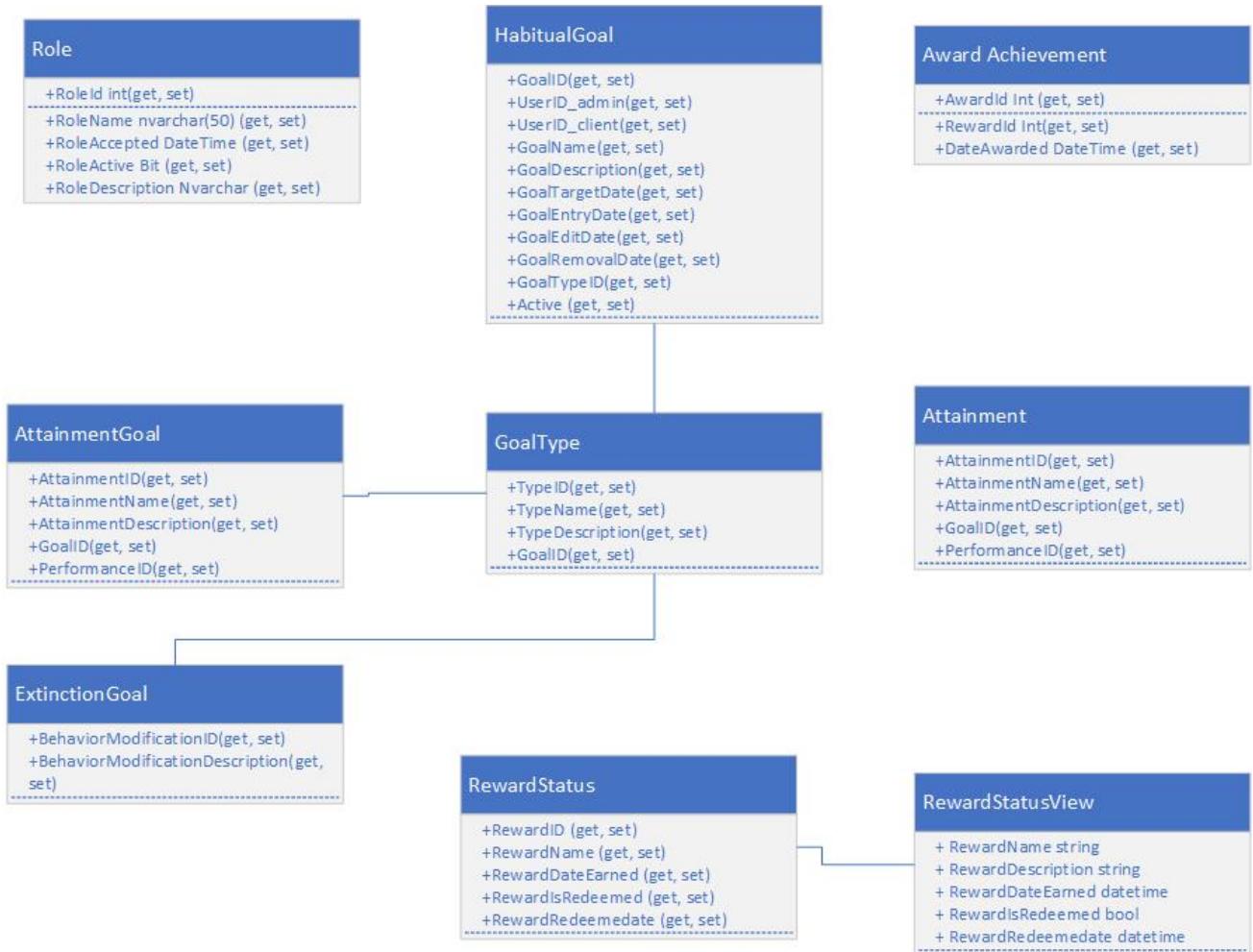
Domain Models

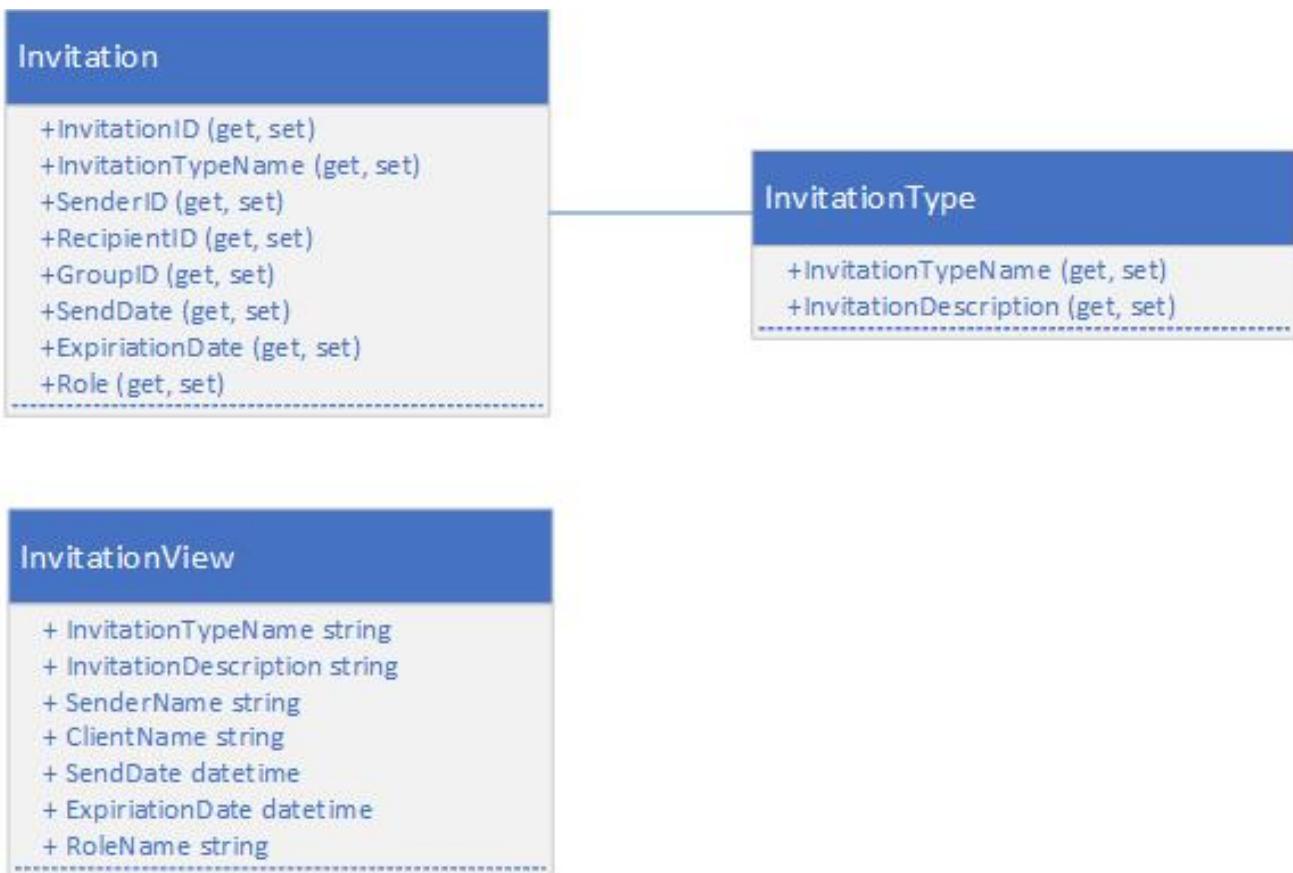
By Nathaniel Webber contributions by: Becky Baenziger, William Clark, Mitchell Paul, Ryan Taylor, Whitney Vinson, Jory Wernette

This section shows the Data and Storage View Models









System Diagrams: Data Flows

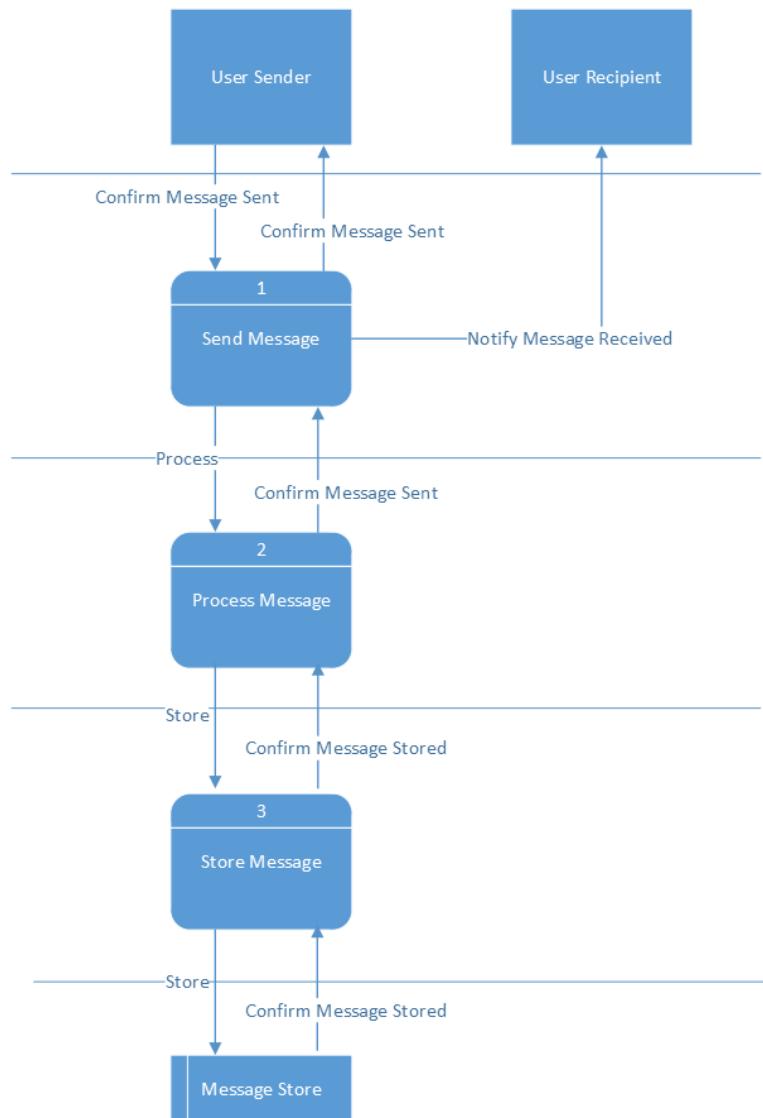
By William Clark contributions by: Becky Baenziger, Mitchell Paul, Ryan Taylor, Whitney Vinson, Nathaniel Webber, Jory Wernette

Messaging System

Messaging

By William Clark

Shows the process of sending an individual message to another user.

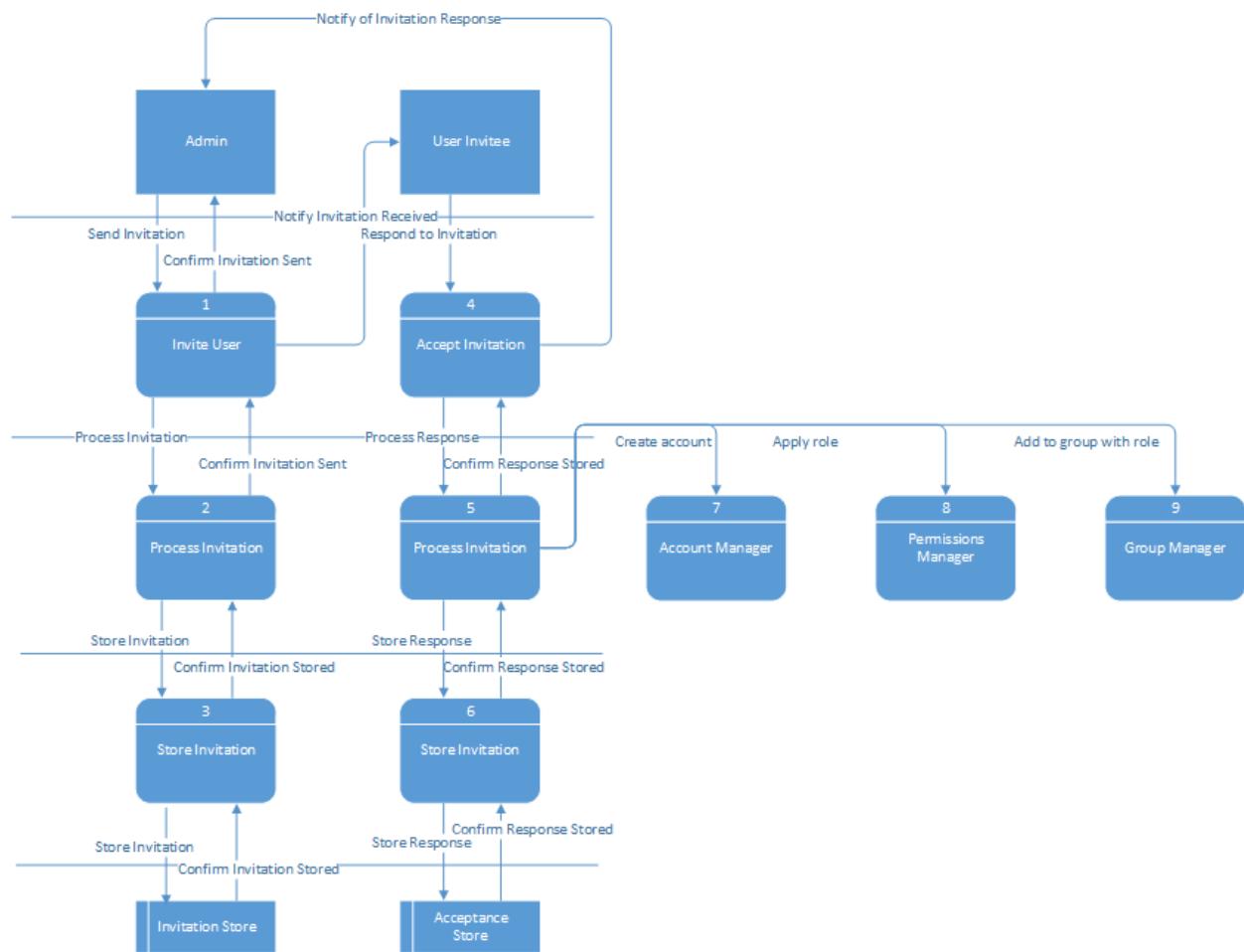


Invitation System

Invitations

By William Clark

Shows the process of sending an invitation to another user. The Process Invitation process manages the implementation of the acceptance of the invitation, sending requests to the various managers to accomplish the purpose of the invitation, depending upon the invitation type.

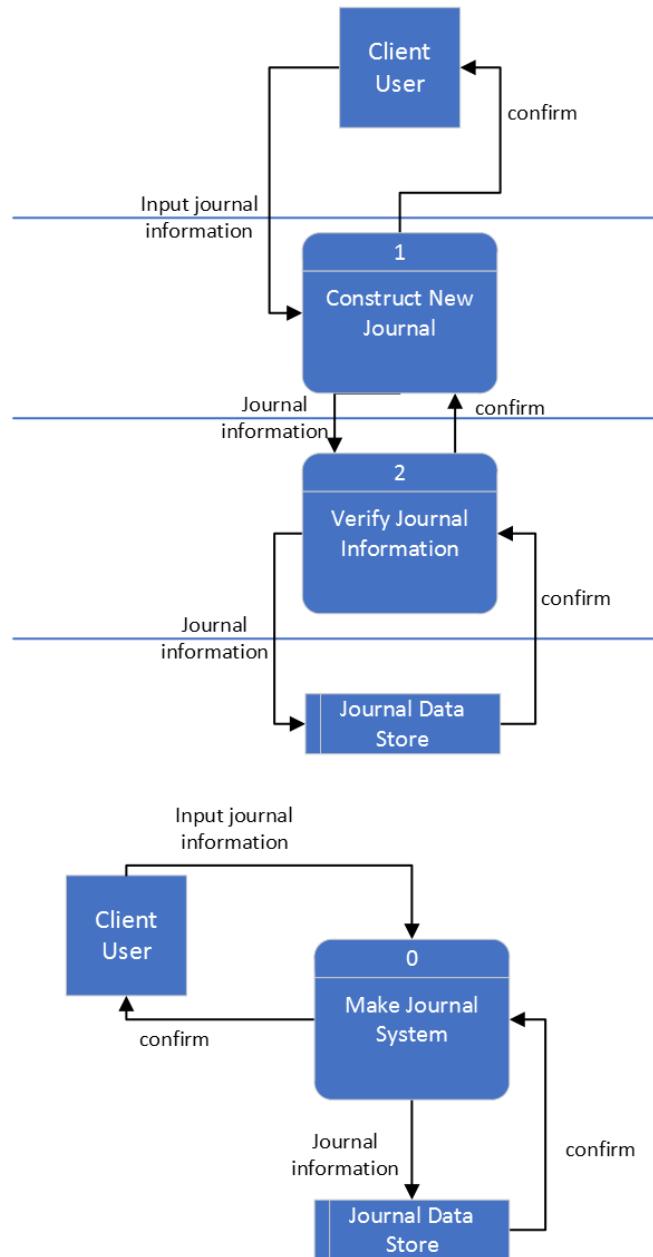


Journal System

Making a Journal

By Ryan Taylor

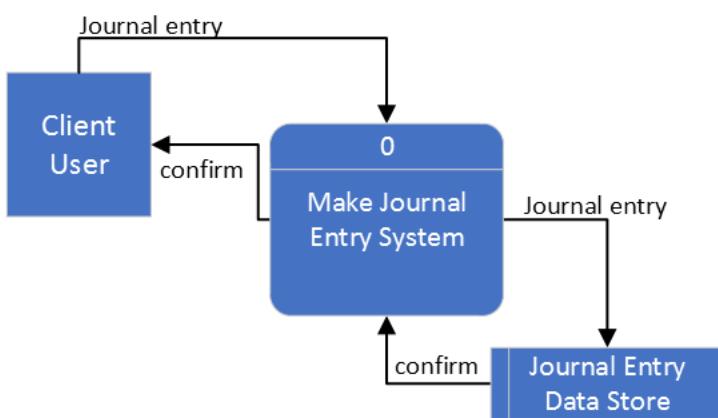
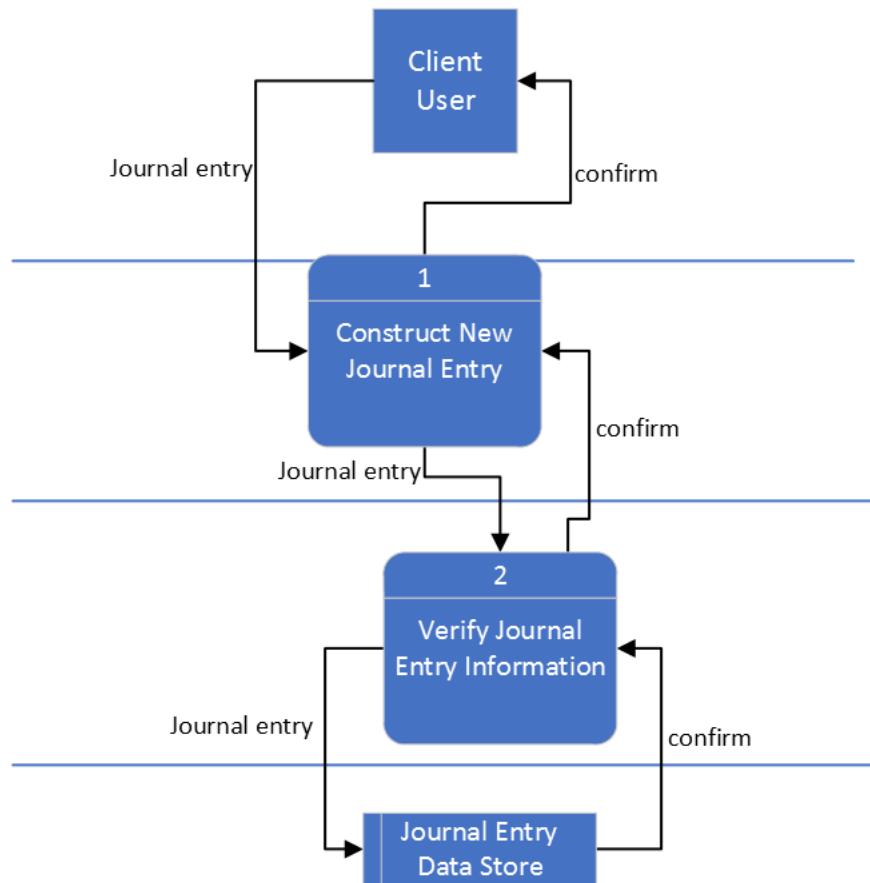
Shows the process of creating a new journal.



Making a Journal Entry

By Ryan Taylor

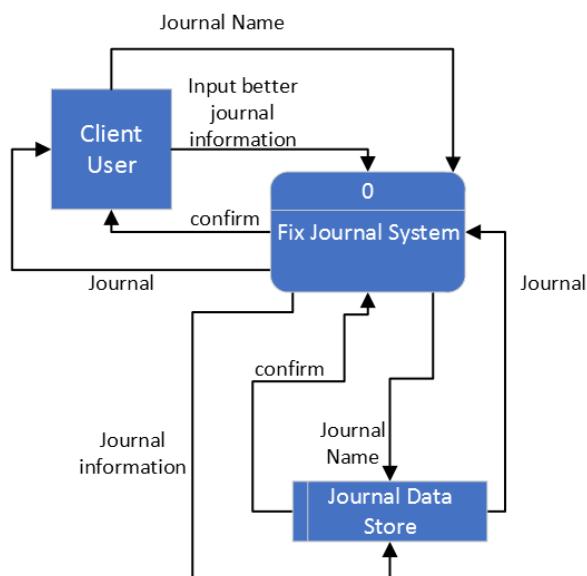
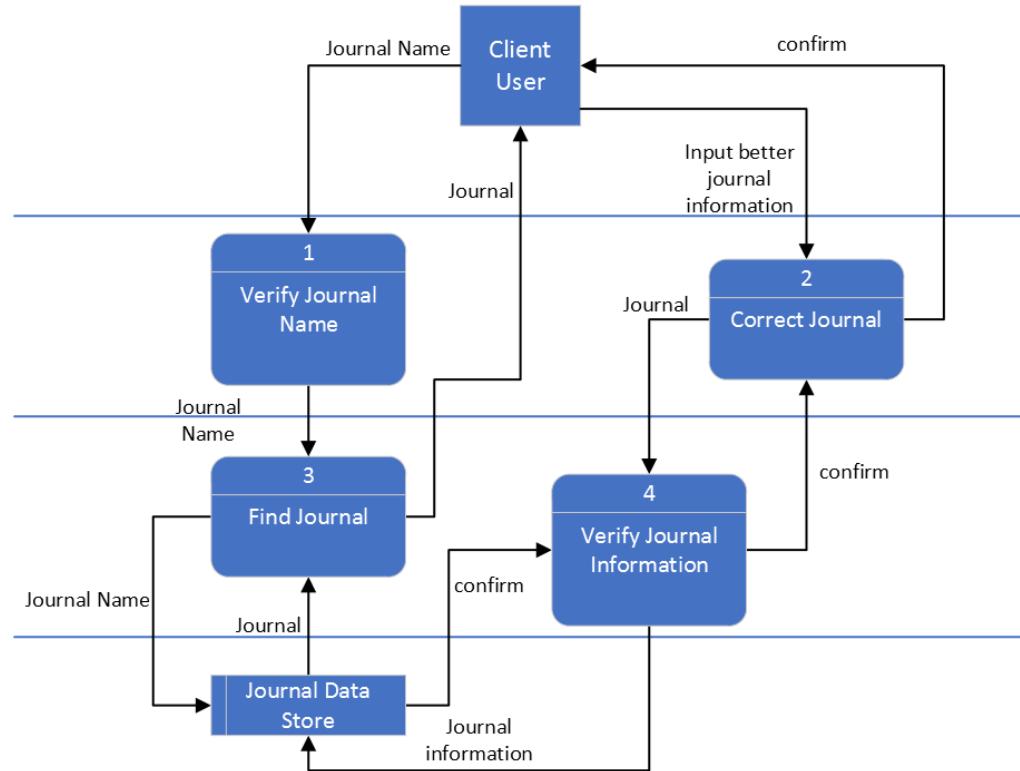
Shows the process of creating a new journal entry in an existing journal.



Editing a Journal

By Ryan Taylor

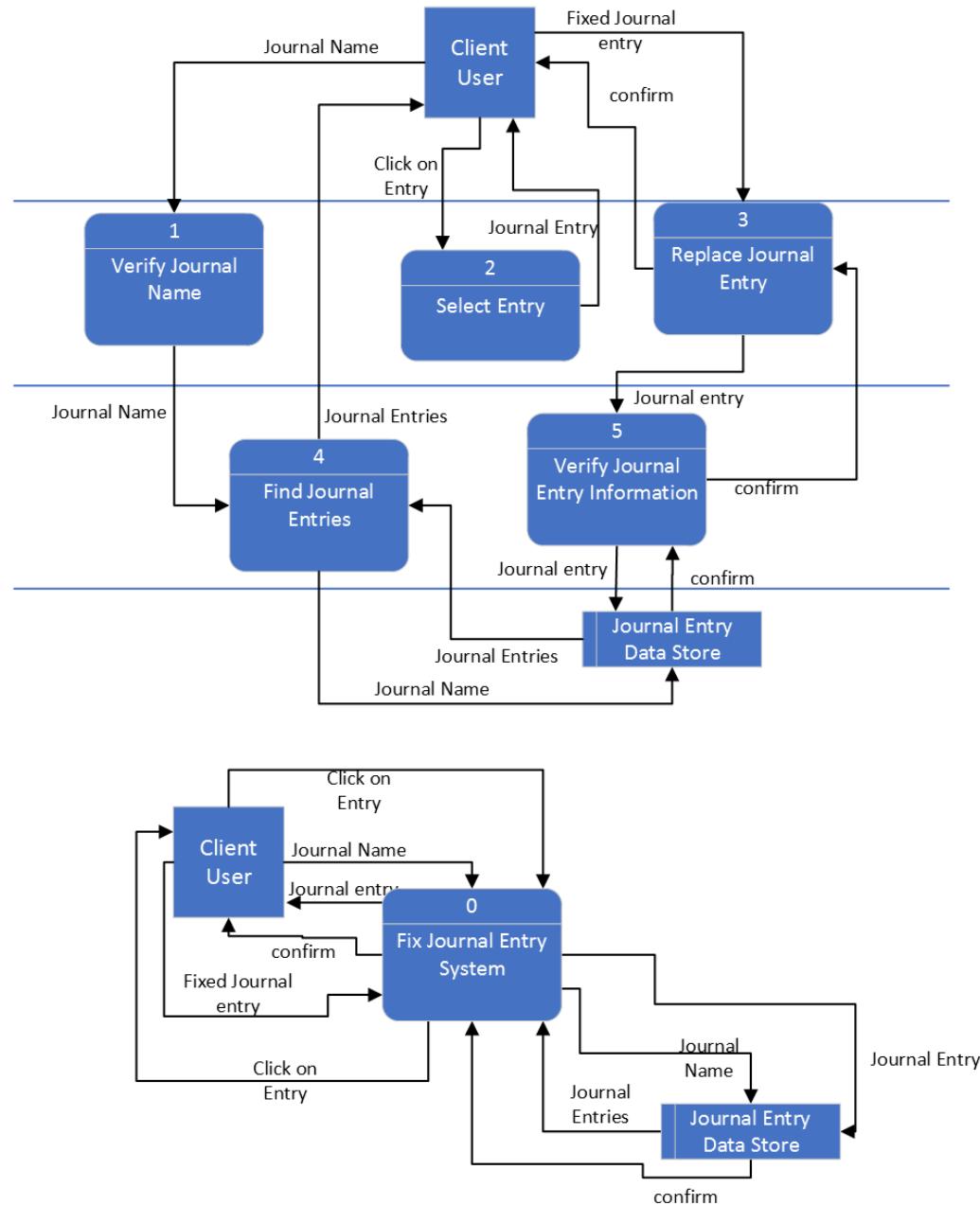
Shows the process of editing an existing journal.



Editing a Journal Entry

By Ryan Taylor

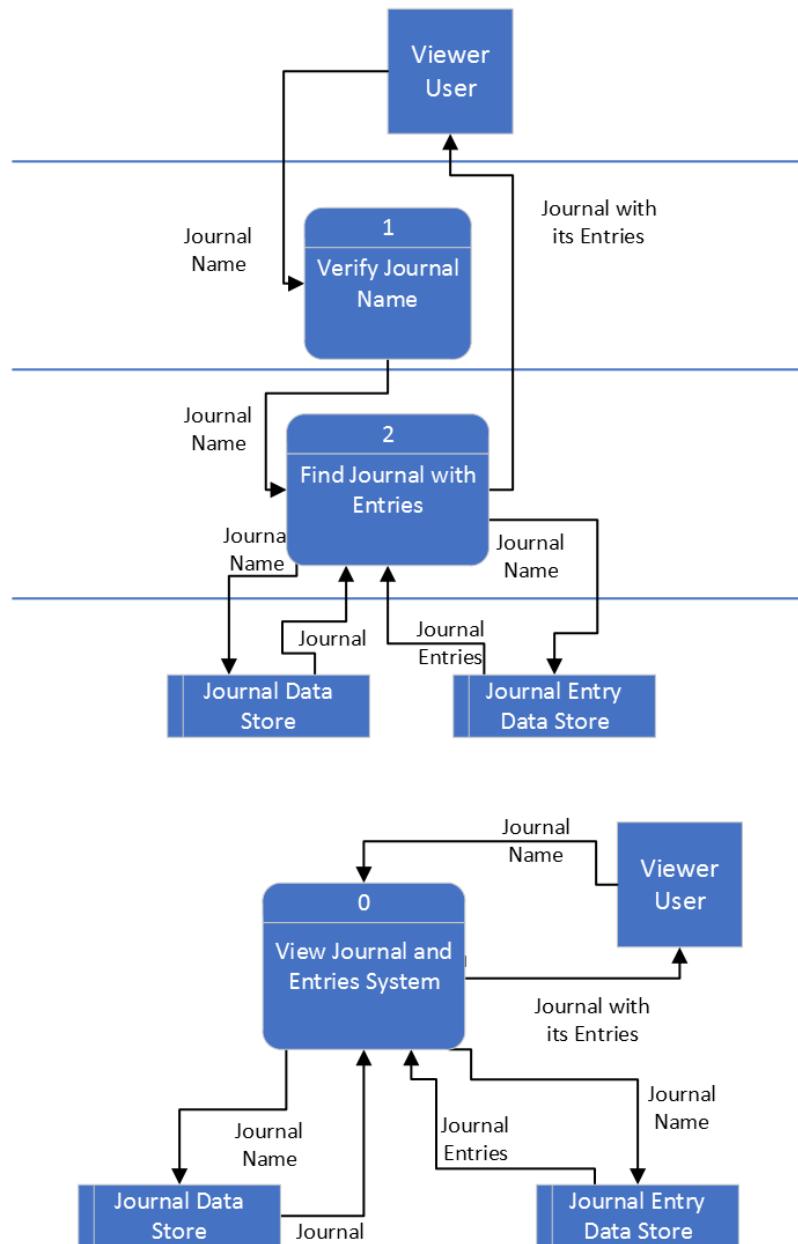
Shows the process of editing an existing journal entry.



Viewing Journals and Entries

By Ryan Taylor

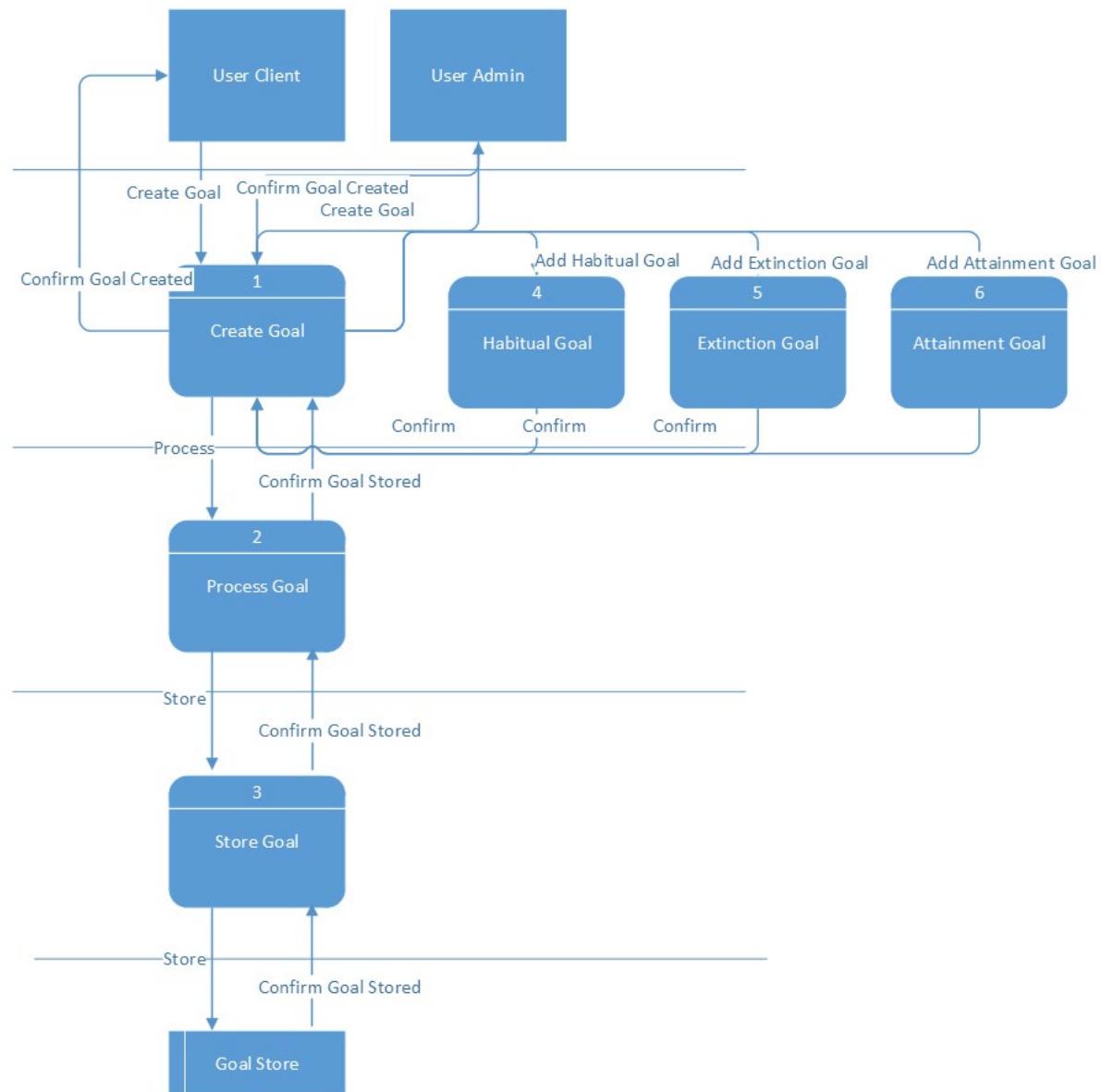
Shows the process of viewing an existing journal and its entries.



Goal System

Create Goal

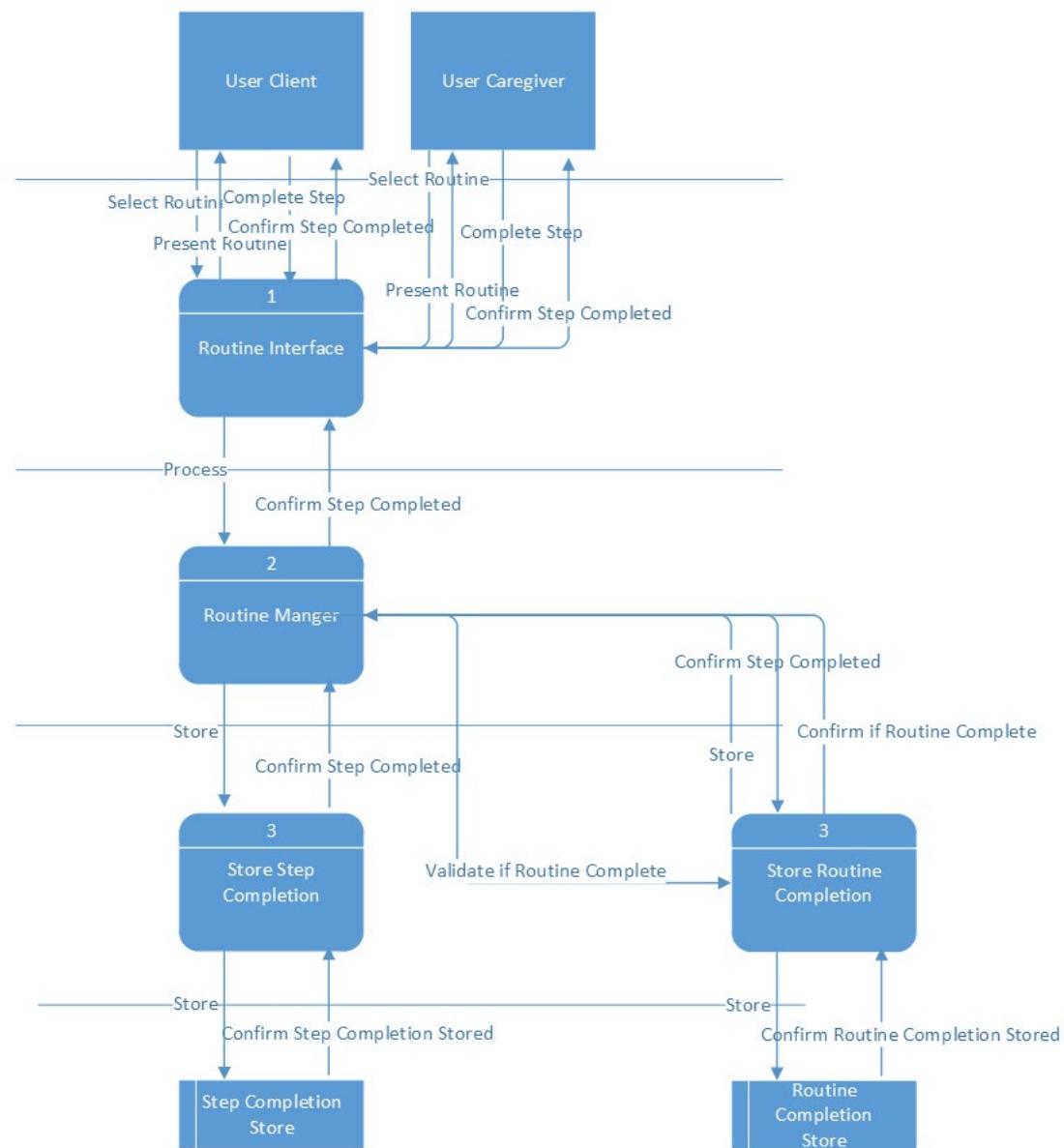
By Becky Baenziger, Mitchel Paul, Nathaniel Webber, Whitney Vinson, William Clark



Routine System

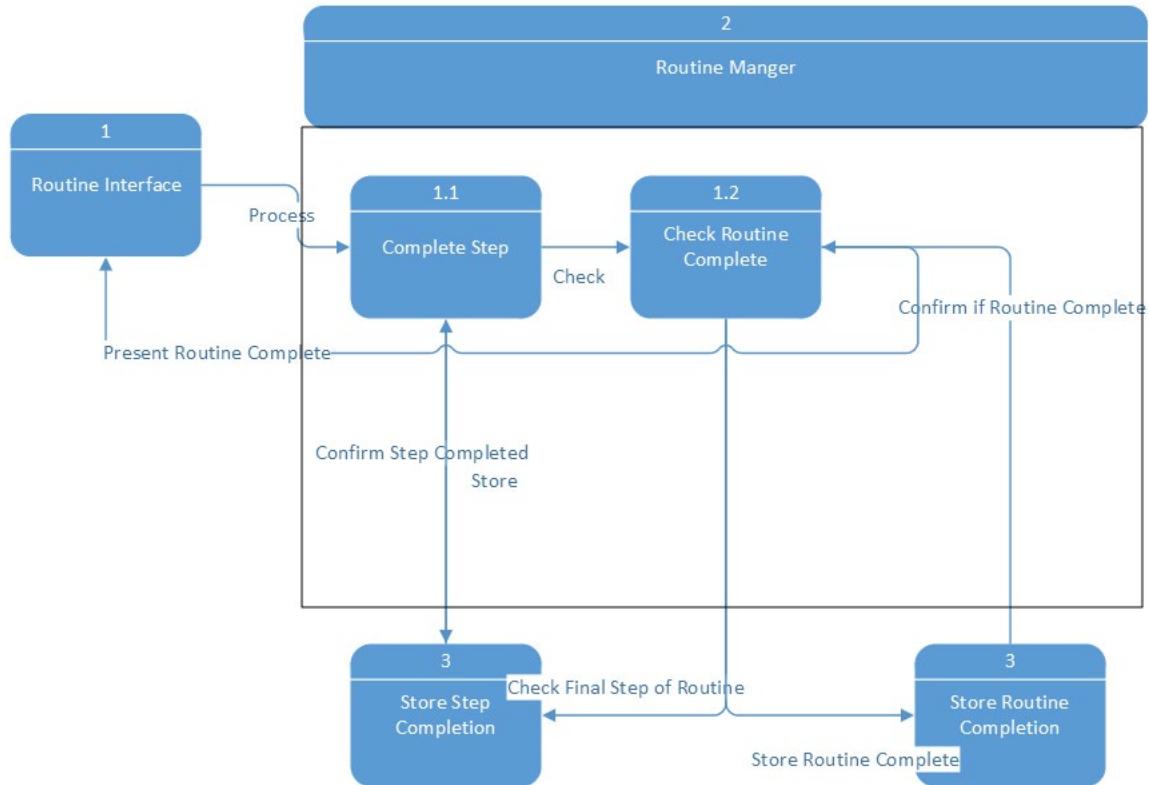
Complete Step/Routine

By William Clark



Complete Step Routine Context Diagram

By William Clark



UML Class Diagrams

By Nathaniel Webber contributions by: Becky Baenziger, William Clark, Mitchell Paul, Ryan Taylor, Whitney Vinson, Jory Wernette

Data Access Layer

The data access layer will handle permanent storage of the input and output from the user to the application.

User

```
<<Interface>>
IUserAccessor

List<User> SelectUserByActive (bool active = true)
List<string> SelectUserByID(int userID)
List<string> SelectAllUsers()
int UpdateUser (User oldUser, User newUser)
int DeactivateUser (int userID)
int ReactivateUser (int userID)
```

Message

```
<<Interface>>
IMessageAccessor

List<Message> SelectMessageBySender (int userID)
List<string> SelectMessageByID(int userID)
List<string> SelectAllMessages()
string UpdateMessage (string messageContent)
int DeactivateMessage (int userID)
int ReactivateMessage (int userID)
```

Membership

```
<<Interface>>
IMembershipAccessor

List<Membership> SelectUserByGroupID (int groupID)
List<Membership> SelectUserByID(int userID)
List<Membership> SelectUserByRoleID(int roleID)
List<string> SelectAllMemberships()
int UpdateMembership (Membership oldMembership,
Membership newMembership)
int DeactivateMembership (int userID)
int ReactivateMembership (int userID)
```

Role

```
<<Interface>>
IRoleAccessor

List<Role> SelectRoleByRoleID (int roleID)
List<string> SelectAllRoles()
int UpdateRole (Role oldRole, Role newRole)
int DeactivateRole (int roleID)
int ReactivateRole (int roleID)
```

Invitation

```
<<Interface>>
IInvitationAccessor

List<Invitation> SelectInvitationByGroupID ( int
groupID)
List<Invitaion> SelectInvitationByUserID(int userID)
List<Invitation> SelectInvitationByRoleID(int roleID)
List<string> SelectAllInvitaion()
int UpdateInvitation ( Invitaion oldInvitaion, Invitation
newInvitation)
int DeactivateInvitaion (int userID)
int ReactivateInvitation (int userID)
```

Journal

```
<<Interface>>
IJournalAccessor

List<Journal> SelectJournalByUserID(int userID)
List<string> SelectAllJournals()
int UpdateJournal (Journal oldJournal, Journal
newJournal)
int DeactivateJournal (int userID)
int ReactivateJournal (int userID)
```

JournalEntry

```
<<Interface>>
IJournalEntryAccessor

List<JournalEntry> SelectJournalEntryByJournalID (int
journalID)
List<JournalEntry> SelectJournalEntryByUserID(int
userID)
List<string> SelectAllJournalEntries()
int UpdateJournalEntry (JournalEntry
oldJournalEntryBody, JournalEntry
newJournalEntryBody)
int DeactivateJournalEntry (int journalID)
int ReactivateJournalEntry (int journalID)
```

Incident

```
<<Interface>>
IIncidentAccessor

List<Incident> SelectIncidentByIncidentID (int
incidentID)
List<Incident> SelectIncidentByUserID(int userID)
List<string> SelectAllIncidents()
int UpdateIncident (Incident oldIncident, Incident
newIncident)
int DeactivateIncident (int incidentID)
int ReactivateIncident (int incidentID)
```

Performance

```
<<Interface>>
IPerformanceAccessor

List<Performance> SelectPerformanceByPerformanceID
(int performanceID)
List<Performance> SelectPerformanceByUserID(int
userID)
List<string> SelectAllPerformance()
int UpdatePerformance (Performance
oldPerformanceDescription, Performance
newPerformanceDescription)
int DeactivatePerformance (int performanceID)
int ReactivatePerformance (int performanceID)
```

AttainmentGoal

```
<<Interface>>
IAttainmentGoalAccessor

List<AttainmentGoal>
SelectAttainmentGoalByAttainmentID (int
attainmentID)
List<AttainmentGoal>
SelectAttainmentGoalByUserID(int userID)
List<AttainmentGoal>
SelectAttainmentGoalByPerformanceID(int
performanceID)
List<string> SelectAllAttainmentGoals()
int UpdateAttainmentGoals (AttainmentGoal
oldAttainmentDescription, AttainmentGoal
newAttainmentDescription)
int DeactivateAttainmentGoal (int attainmentID)
int ReactivateAttainmentGoal (int attainmentID)
```

ExtinctionGoal

```
<<Interface>>
IExtinctionGoalAccessor

List<ExtinctionGoal>
SelectExtinctionGoalByExtinctionID (int extinctionID)
List<ExtinctionGoal> SelectExtinctionGoalByUserID(int
userID)
List<ExtinctionGoal>
SelectExtinctionGoalByPerformanceID(int
performanceID)
List<string> SelectAllExtinctionGoals()
int UpdateExtinctionGoals (ExtinctionGoal
oldExtinctionDescription, ExtinctionGoal
newExtinctionDescription)
int DeactivateExtinctionGoal (int extinctionID)
int ReactivateExtinctionGoal (int extinctionID)
```

HabitualGoal

```
<<Interface>>
IHabitualGoalAccessor

List<HabitualGoal> SelectHabitualGoalByHabitualID
(int habitualID)
List<HabitualGoal> SelectHabitualGoalByUserID(int
userID)
List<HabitualGoal>
SelectHabitualGoalByPerformanceID(int
performanceID)
List<string> SelectAllHabitualGoals()
int UpdateHabitualGoals (HabitualGoal
oldHabitualDescription, HabitualGoal
newHabitualDescription)
int DeactivateHabitualGoal (int habitualID)
int ReactivateHabitualGoal (int habitualID)
```

GoalType

```
<<Interface>>
IGoalTypeAccessor

List<GoalType> SelectGoalTypeByGoalTypeID (int
goalTypeID)
List<GoalType> SelectGoalTypeByGoalID(int goalID)
List<string> SelectAllAttainmentGoalTypes()
int UpdateGoalType (GoalType
oldGoalTypeDescription, GoalType
newGoalTypeDescription)
int DeactivateGoalType (int goalTypeID)
int ReactivateGoalType (int goalTypeID)
```

Routine

```
<<Interface>>
IRoutineAccessor

List<Routine> SelectRoutineByRoutineID (int routineID)
List<Routine> SelectRoutineByUserID(int userID)
List<Routine> SelectRoutineByActive(bool active = true)
List<string> SelectAllRoutines()
int UpdateRoutine ( Routine oldRoutineDescription,
Routine newRoutineDescription)
int DeactivateRoutine (int routineID)
int ReactivateRoutine (int routineID)
```

RoutineStep

```
<<Interface>>
IRoutineStepAccessor

List<RoutineStep> SelectRoutineStepByRoutineStepID (int
routineStepID)
List<RoutineStep> SelectRoutineStepByRoutineID(int
RoutineID)
List<RoutineStep> SelectRoutineStepByActive(bool active
= true)
List<string> SelectAllRoutineSteps()
int UpdateRoutineStep ( RoutineStep
oldRoutineStepDescription, RoutineStep
newRoutineStepDescription)
int DeactivateRoutineStep (int routineStepID)
int ReactivateRoutineStep (int routineStepID)
```

RoutineStepCompletion

```
<<Interface>>
IRoutineStepCompletionAccessor

List<RoutineStepCompletion>
SelectRoutineStepCompletionByRoutineStepID (int routineStepID)
List<RoutineStepCompletion>
SelectRoutineStepCompletionByUserID(int userID)
List<string> SelectAllRoutineStepCompletion()
int UpdateRoutineStepCompletion (
RoutineStepCompletion oldRoutineStepDescription,
RoutineStepCompletion newRoutineStepDescription)
int DeactivateRoutineStepCompletion (int routineStepID)
int ReactivateRoutineStepCompletion (int routineStepID)
```

RoutineCompletion

```
<<Interface>>
IRoutineCompletionAccessor

List<RoutineCompletion>
SelectRoutineCompletionByRoutineID (int routineID)
List<RoutineCompletion>
SelectRoutineCompletionByUserID(int userID)
List<string> SelectAllRoutineCompletion()
int UpdateRoutineCompletion ( RoutineCompletion
oldRoutineDescription, RoutineCompletion
newRoutineDescription)
int DeactivateRoutineCompletion (int routineID)
int ReactivateRoutineCompletion (int routineID)
```

Award

```
<<Interface>>
IAwardAccessor

List<Award> SelectAwardByAwardID (int awardID)
List<Award> SelectAwardByGoalID(int goalID)
List<Award> SelectAwardByGoalTypeID( int goalTypeID)
List<string> SelectAllAwards()
int UpdateAwards ( Award oldAwardDescription, Award newAwardDescription)
int DeactivateAward (int awardID)
int ReactivateAward (int awardID)
```

Reward

```
<<Interface>>
IRewardAccessor

List<Reward> SelectRewardStepByRewardID (int rewardID)
List<Reward> SelectRewardStepByUserID(int userID)
List<string> SelectAllRewards()
int UpdateReward ( Reward oldRewardDescription,
Reward newRewardDescription)
int DeactivateReward (int rewardID)
int ReactivateReward (int rewardID)
```

Logic Layer

The logic layer will handle the computations, calculations, processing, and traffic management of the application.

User

UserManager

```
RetrieveUserByActive (bool active = true)
RetrieveUserByID(int userID)
RetrieveAllUsers()
AddNewUser(User firstName, User lastName,
User email, User password)
EditUser (User oldUser, User newUser)
RemoveUser (int userID)
```

Message

MessageManager

```
RetrieveMessageBySender (int userID)
RetrieveMessageByUserID(int userID)
RetrieveAllMessages()
AddNewMessage(Invitation senderID userID, Invitation
recipientID, Message messageBody)
EditMessage (string messageContent)
RemoveMessage (int userID)
```

Membership

MembershipManager

```
RetrieveUserByGroupID (int groupID)
RetrieveUserByID(int userID)
RetrieveUserByRoleID( int roleID)
RetrieveAllMemberships()
AddNewMembership(Membership userID,
Membership groupID)
EditMembership (Membership oldMembership,
Membership newMembership)
RemoveMembership (int userID)
```

Role

RoleManager

```
RetrieveRoleByRoleID (int roleID)
RetrieveAllRoles()
AddNewRole(Role roleName, Role roleDescription)
EditRole (Role oldRole, Role newRole)
RemoveRole (int roleID)
```

Invitation

InvitationManager

```
RetrieveInvitationByGroupID (int groupID)
RetrieveInvitationByUserID(int userID)
RetrieveInvitationByRoleID(int roleID)
RetrieveAllInvitaion()
AddNewInvitation(Invitation senderID, Invitation
recipientID, Invitation invitationTypeName)
EditInvitation ( Invitaion oldInvitaion, Invitation
newInvitation)
RemoveInvitaion (int userID)
```

Journal

JournalManager

```
RetrieveJournalByUserID(int userID)
RetrieveAllJournals()
AddNewJournal(Journal journalName)
EditJournal (Journal oldJournal, Journal newJournal)
RemoveJournal (int userID)
```

JournalEntry

JournalEntryManager

```
RetrieveJournalEntryByJournalID (int journalID)
RetrieveJournalEntryByUserID(int userID)
RetrieveAllJournalEntries()
AddNewJournalEntry(JournalEntry journalName,
JournalEntry entryBody)
EditJournalEntry (JournalEntry oldJournalEntryBody,
JournalEntry newJournalEntryBody)
RemoveJournalEntry (int journalID)
```

Incident

IncidentManager

```
RetrieveIncidentByIncidentID (int incidentID)
RetrieveIncidentByUserID(int userID)
RetrieveAllIncidents()
AddNewIncident(Incident incidentDate, int userID,
Incident incidentBody)
EditIncident (Incident oldIncident, Incident
newIncident)
RemoveIncident (int incidentID)
```

Performance

PerformanceManager

```
RetrievePerformanceByPerformanceID (int performanceID)
RetrievePerformanceByUserID(int userID)
RetrieveAllPerformance()
AddNewPerformance(Performance performanceName
Performance performanceDescription)
EditPerformance (Performance
oldPerformanceDescription, Performance
newPerformanceDescription)
RemovePerformance (int performanceID)
```

AttainmentGoal

AttainmentGoalManager

```
RetrieveAttainmentGoalByAttainmentID (int
attainmentID)
RetrieveAttainmentGoalByUserID(int userID)
RetriveAttainmentGoalByPerformanceID(int
performanceID)
RetriveAllAttainmentGoals()
AddNewAttainmentGoal(GoalType goalTypeName,
GoalType goalTypeDescription)
EditAttainmentGoals (AttainmentGoal
oldAttainmentDescription, AttainmentGoal
newAttainmentDescription)
RemoveAttainmentGoal (int attainmentID)
```

ExtinctionGoal

ExtinctionGoalManager

```
RetriveExtinctionGoalByExtinctionID (int  
attainmentID)  
RetriveExtinctionGoalByUserID(int userID)  
RetriveExtinctionGoalByPerformanceID(int  
performanceID)  
RetrieveAllExtinctionGoals()  
AddNewExtinctionGoal(GoalType goalTypeName,  
GoalType goalDescription)  
EditExtinctionGoals (ExtinctionGoal  
oldExtinctionDescription, ExtinctionGoal  
newExtinctionDescription)  
RemoveExtinctionGoal (int extinctionID)
```

HabitualGoal

HabitualGoalManager

```
RetriveHabitualGoalByHabitualID (int habitualID)  
RetriveHabitualGoalByUserID(int userID)  
RetriveHabitualGoalByPerformanceID(int  
performanceID)  
RetriveAllHabitualGoals()  
AddNewHabitualGoal(GoalType goalTypeName,  
GoalType goalDescription)  
EditHabitualGoals (HabitualGoal  
oldHabitualDescription, HabitualGoal  
newHabitualDescription)  
RemoveHabitualGoal (int habitualID)
```

GoalType

GoalTypeManager

```
RetriveGoalTypeByGoalTypeID (int goalTypeID)  
RetriveGoalTypeByGoalID(int goalID)  
RetriveAllGoalTypes()  
EditGoalType (GoalType oldGoalTypeDescription,  
GoalType newGoalTypeDescription)  
RemoveGoalType (int goalTypeID)
```

Routine

RoutineManager

```
RetriveRoutineByRoutineID (int routineID)
RetriveRoutineByUserID(int userID)
RetriveRoutineByActive(bool active = true)
RetriveAllRoutines()
AddNewRoutine(Routine routineName, Routine
routineSetter, Routine routineStartDate, Routine
routineEndDate, Routine goalName)
EditRoutine ( Routine oldRoutineDescription, Routine
newRoutineDescription)
RemoveRoutine (int routineID)
```

RoutineStep

RoutineStepManager

```
RetriveRoutineStepByRoutineStepID (int routineStepID)
RetriveRoutineStepByRoutineID(int RoutineID)
RetriveRoutineStepByActive(bool active = true)
RetriveAllRoutineSteps()
AdnewRoutineStep(RoutineStep routineName,
EditRoutineStep ( RoutineStep oldRoutineStepDescription,
RoutineStep newRoutineStepDescription)
RemoveRoutineStep (int routineStepID)
```

RoutineStepCompletion

RoutineStepCompletionManager

```
RetriveRoutineStepByRoutineStepID (int routineStepID)
RetriveRoutineStepByUserID(int userID)
RetriveAllRoutineStepCompletion()
AddRoutineStep(RoutineStep routineName, RoutineStep
routineStepName, RoutineStep routineStartDate,
RoutineStep routineCompletionDate)
EditRoutineStepCompletion ( RoutineStepCompletion
oldRoutineStepDescription, RoutineStepCompletion
newRoutineStepDescription)
RemoveRoutineStepCompletion (int routineStepID)
```

RoutineCompletion

RoutineCompletionManager

```
RetrieveRoutineCompletionByRoutineID (int routineID)
RetrieveRoutineCompletionByUserID(int userID)
RetrieveAllRoutineCompletion()
EditRoutineCompletion ( RoutineCompletion
oldRoutineDescription, RoutineCompletion
newRoutineDescription)
RemoveRoutineCompletion (int routine ID)
```

Award

AwardManager

```
RetrieveAwardByAwardID ( int awardID)
RetrieveAwardByGoalID(int goalID)
RetrieveAwardByGoalTypeID( int goalTypeID)
RetrieveAllAwards()
AddNewAward(Award awardName, Award
awardDescription, Award awardTypeName)
EditAwards ( Award oldAwardDescription, Award
newAwardDescription)
RemoveAward (int awardID)
```

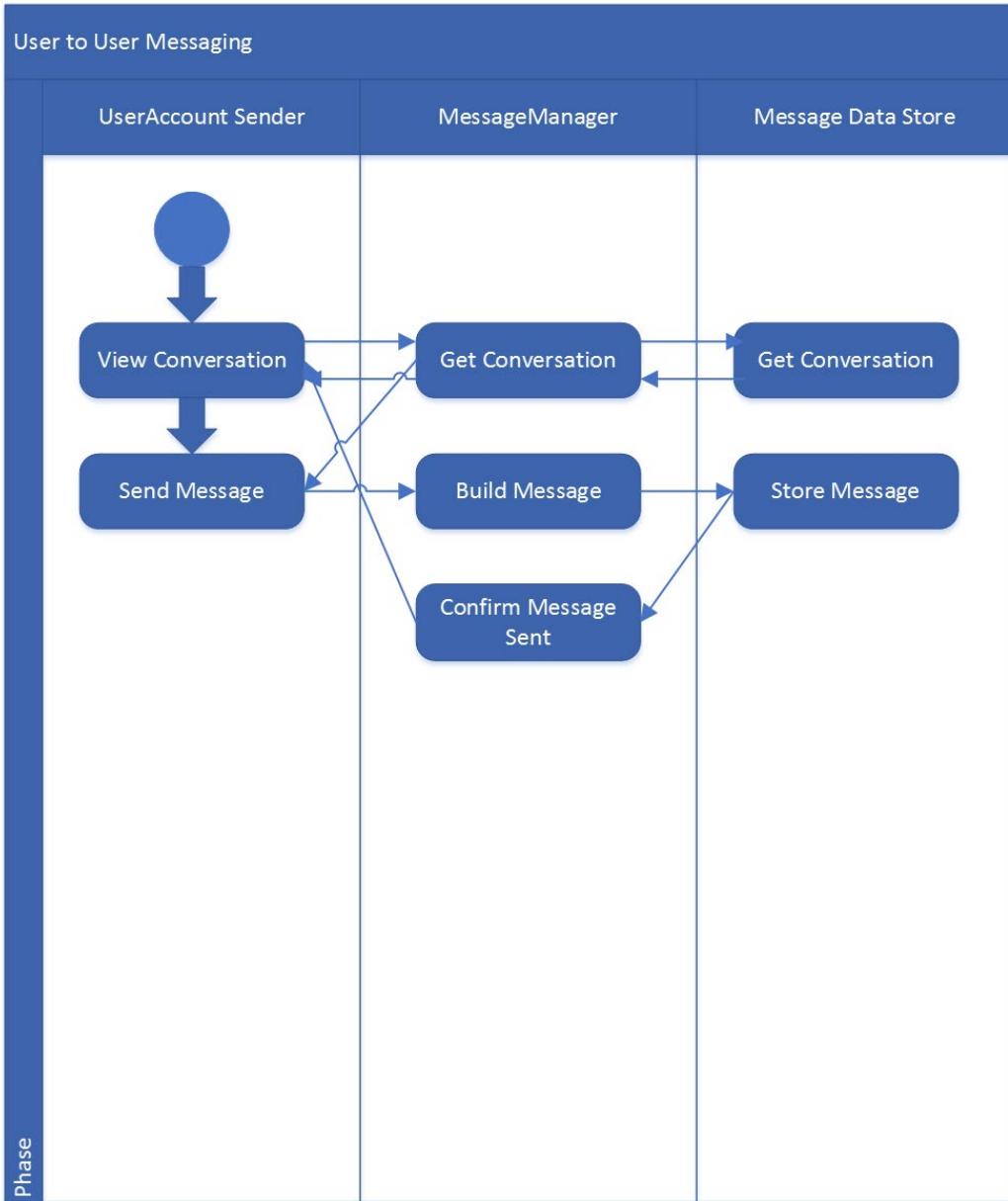
Reward

RewardManager

```
RetrieveRewardStepByRewardID ( int rewardID)
RetrieveRewardStepByUserID(int userID)
RetrieveAllRewards()
AddNewReward(Reward rewardName, Reward
rewardDescription, Reward rewardPrize)
EditReward ( Reward oldRewardDescription, Reward
newRewardDescription)
RemoveReward (int rewardID)
```

Additional Processing Models

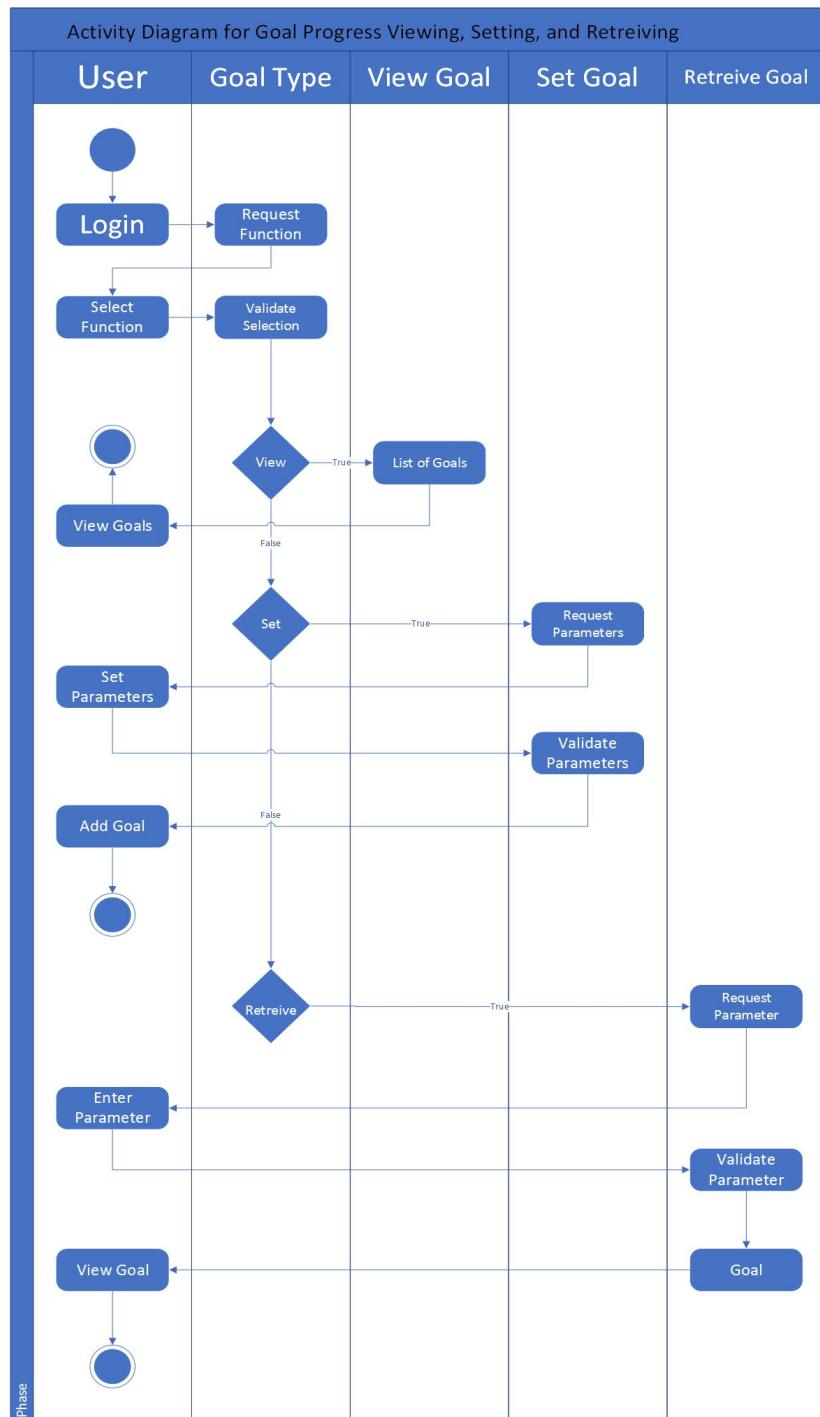
Activity Diagrams



User to User Messaging

By William Clark

Shows the process of user-to-user messaging, and interaction between layers.



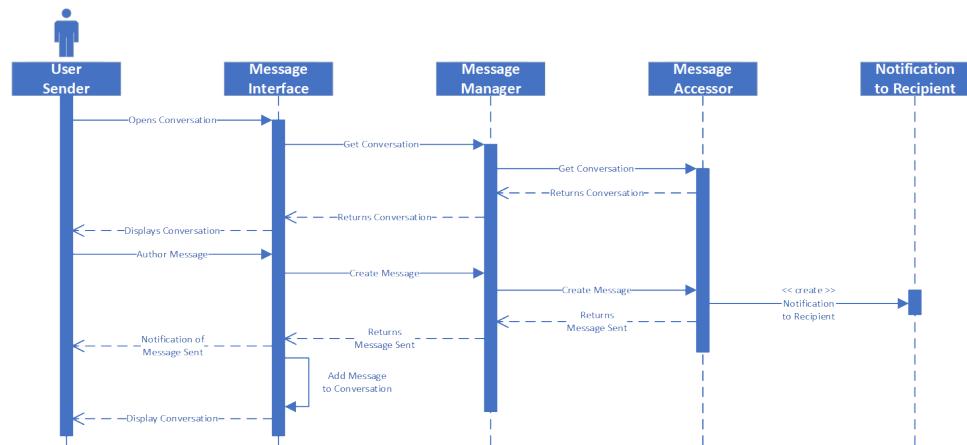
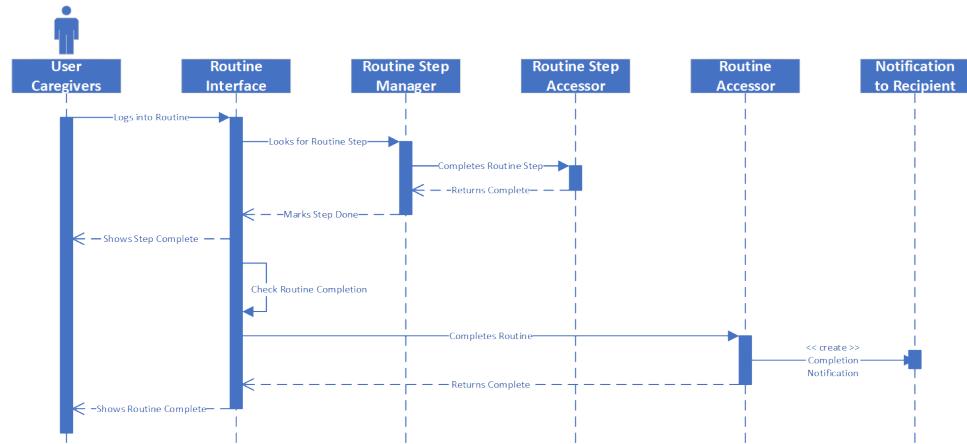
Goal progress viewing, setting, and receiving

By Nathaniel Webber

Depicts the interaction between data objects for the process of viewing, setting, and receiving goals.

Sequence Diagrams

By Whitney Vinson, William Clark



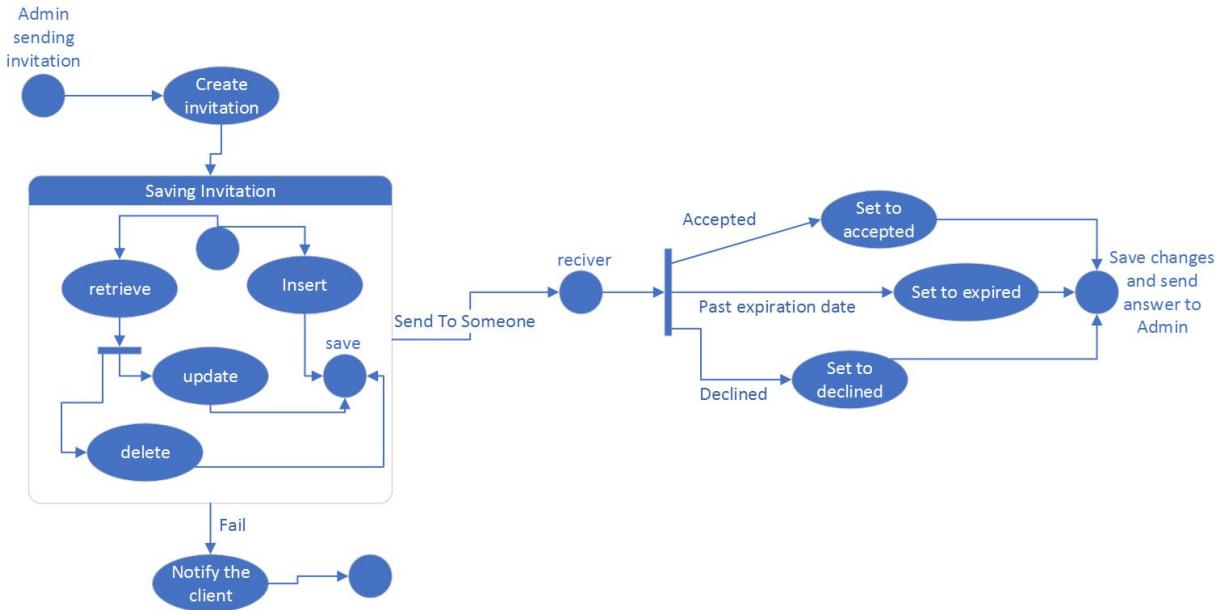
State Chart Diagrams

By Becky Baenziger and Whitney Vinson

Invitations

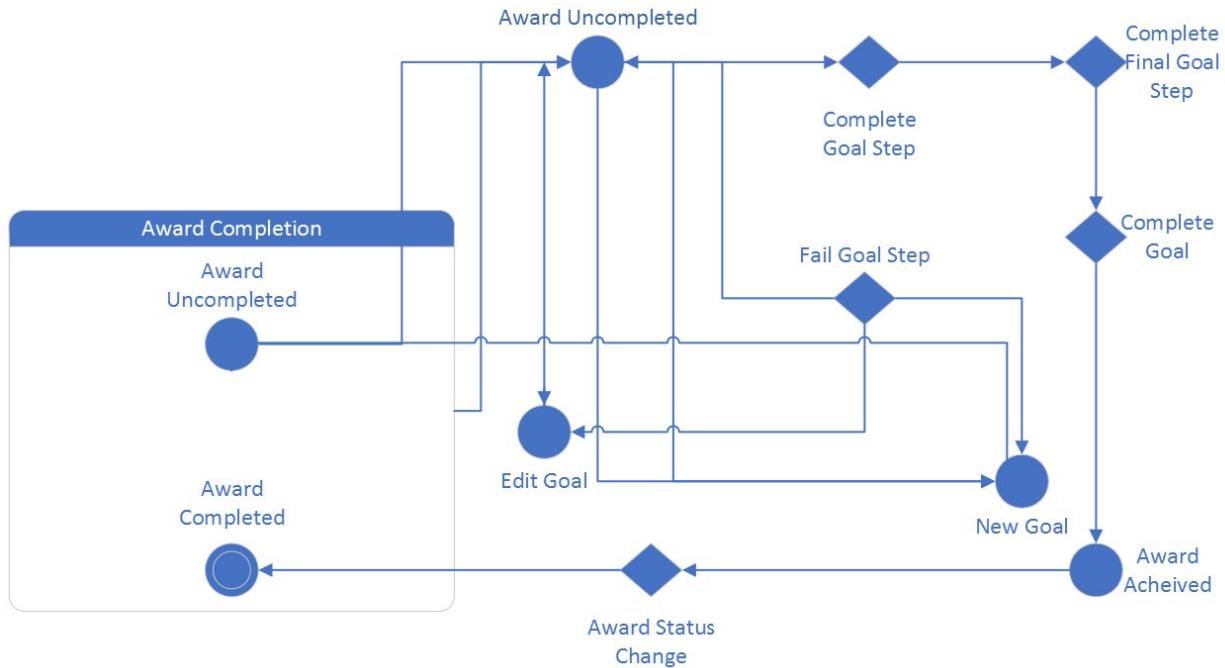
By Whitney Vinson

Details the system state during the process of sending an invitation.



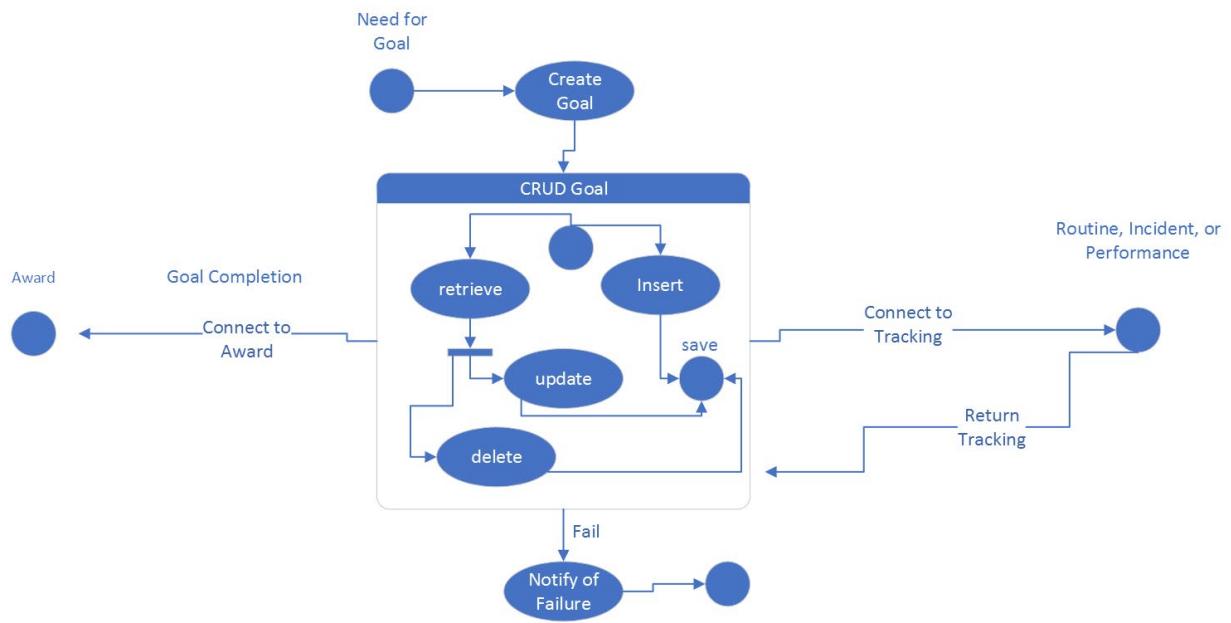
Awards

By Mitchell Paul, edited by Becky Baenziger and Jory Wernette



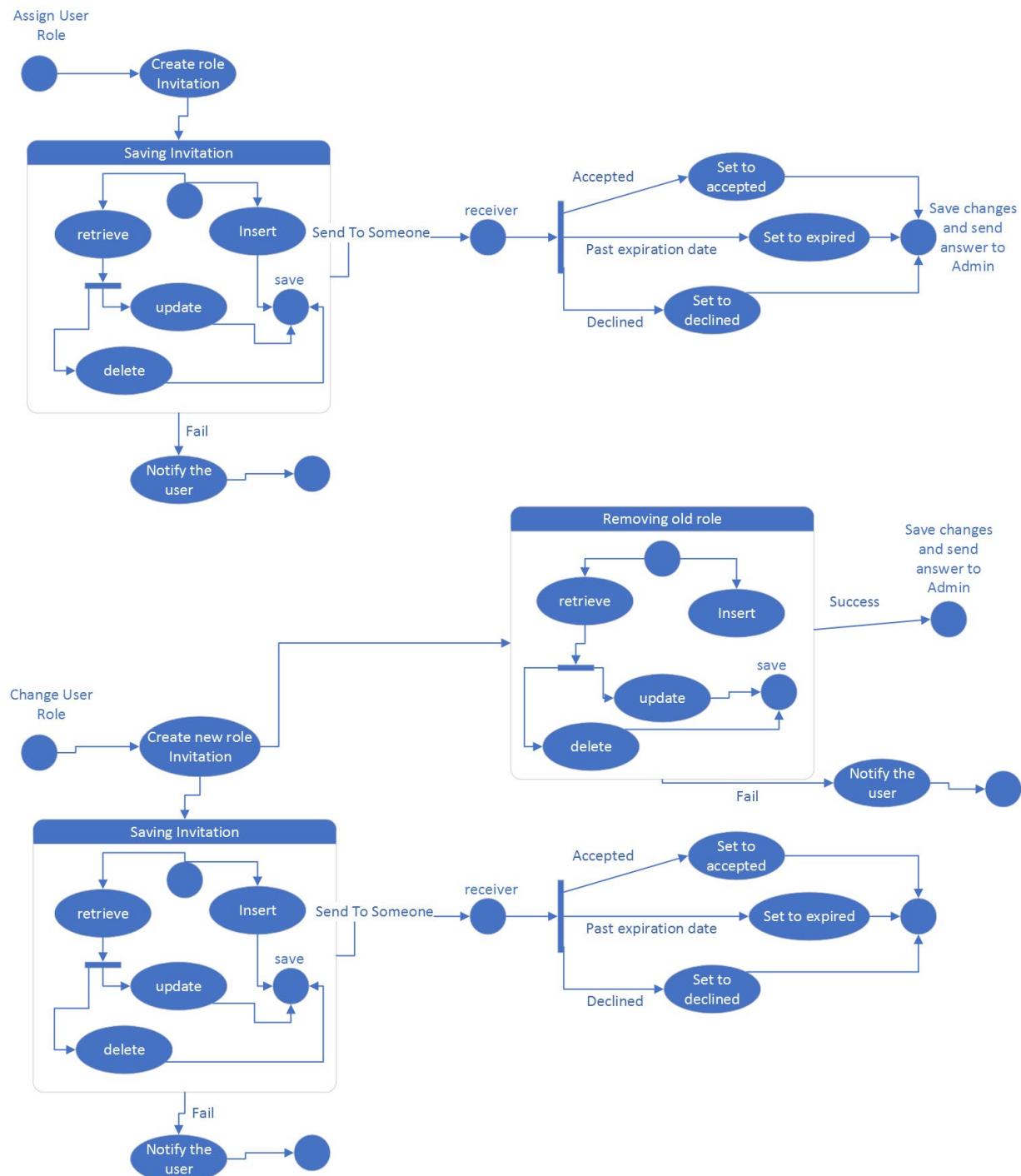
Goals

By Becky Baenziger



Assign/Change User Roles

By Jory Wernette



Deployment Plan

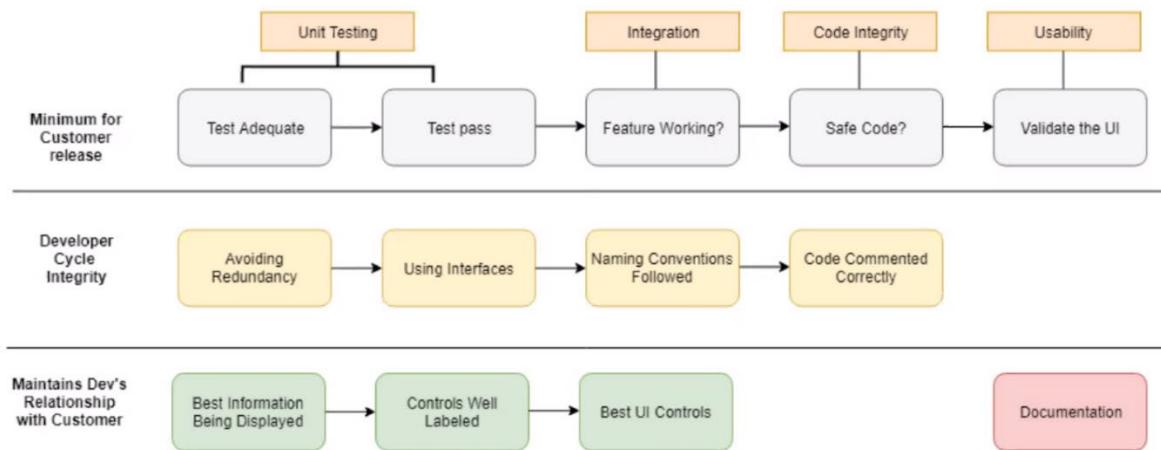
By Whitney Vinson

Project Methodology

During our development of this project, we will be using the Scrum product management workflow and Agile methodologies. To do this, we will be using Workspaces and the GitHub Burn Down Feature to manage the backlog enhanced by project planning tools provided by Zen Hub.

A project manager will be designated to manage the project. We will use four three-week sprints, with a planning week. Code Review will take place during planning weeks.

QA Standards



The product owner will be responsible for the back log, sprint goals and QA. The product will have a three-phase QA process.

Phase One: Unit test are adequate and pass; Features are working; Exceptions are handled; UI is validated;

Phase Two: Code review; (During planning week)

Phase Three: Acceptance testing and documentation review.

Development Technologies

By William Clark

Technology	Version	Use
.NET Framework	4.7.2	Application development framework.
C#	7.3	Primary programming language for application logic.
TSQL	14.00.2027	Database scripting language.
Microsoft SQL Server	2019	Database server software.
Windows Presentation Foundation		Desktop application presentation layer development.
XAML		Desktop application presentation layer development.
Microsoft Visual Studio	2019 Enterprise	Integrated Development Environment.
ECMAScript	2018	Web Presentation layer scripting language.
Bootstrap	4.5.X	Web presentation layer toolkit.
jQuery	3.5.1	Web presentation layer library.
jQuery Validate	1.19.2	Web presentation layer library.
jQuery Mobile	1.4.5	Web presentation layer library.
HTML	5	Web presentation layer markup language.
CSS	4	Web presentation layer style scripting language.

Target Deployment Technologies

By William Clark

Technology	Version	Use
Chrome, Firefox, Safari, and Edge	2020 and above	Web browser.
Windows 10	10 April 2018 Update (Version 1803) and above	Desktop environment.

Programming Team

Becky Baenziger

William Clark

Nick Loesel

Mitchell Paul

Ryan Taylor

Whitney Vinson

Nathaniel Webber

Jory Wernette