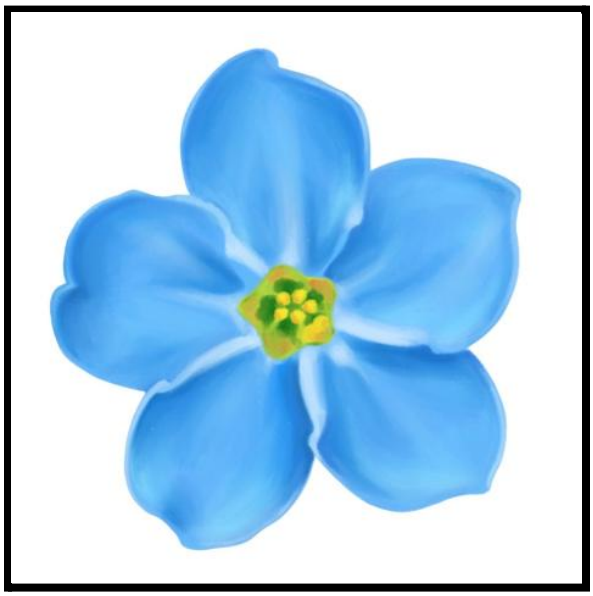
CHERISH



Project Team:

Ke’Shawn Posey

Patrick Ksor

Eric Yan

Srushti Honnenahalli

Table of Contents

---

1. Project Definition

- Why (it is needed)

This project is needed because more often than not, people get so lost in their own professional lives that they forget about their responsibilities. This project is a way to help with that by providing a convenient way to remind someone that they need to contact someone dear to them every once in a while.

- What (is the goal of the project)

The goal of this project is to develop a mobile application that reminds the user to call or text their contacts to who they have not reached out in a while. The application will be able to save contacts either manually or synced automatically, it will also be able to create user accounts to save the user's information and preferences.

- How (how will it be achieved)

This project will be achieved by working in a four-person team and dividing the work amongst them to make sure that the project is completed and functional by the deadline.

2. Project Requirements

- Functional - local database, notification sending, account creation with email or phone, importing contacts.

- Usability
    - User interface - simple user-friendly interface, with the basic functionality to let the application fulfill its purpose.
      Front end software: Android Studio.
    - Performance - No delays are expected as the app only utilizes relatively simple processes, as a result, no real loading times are expected.

Another note to make regarding this app is that it will more often be running in the background.

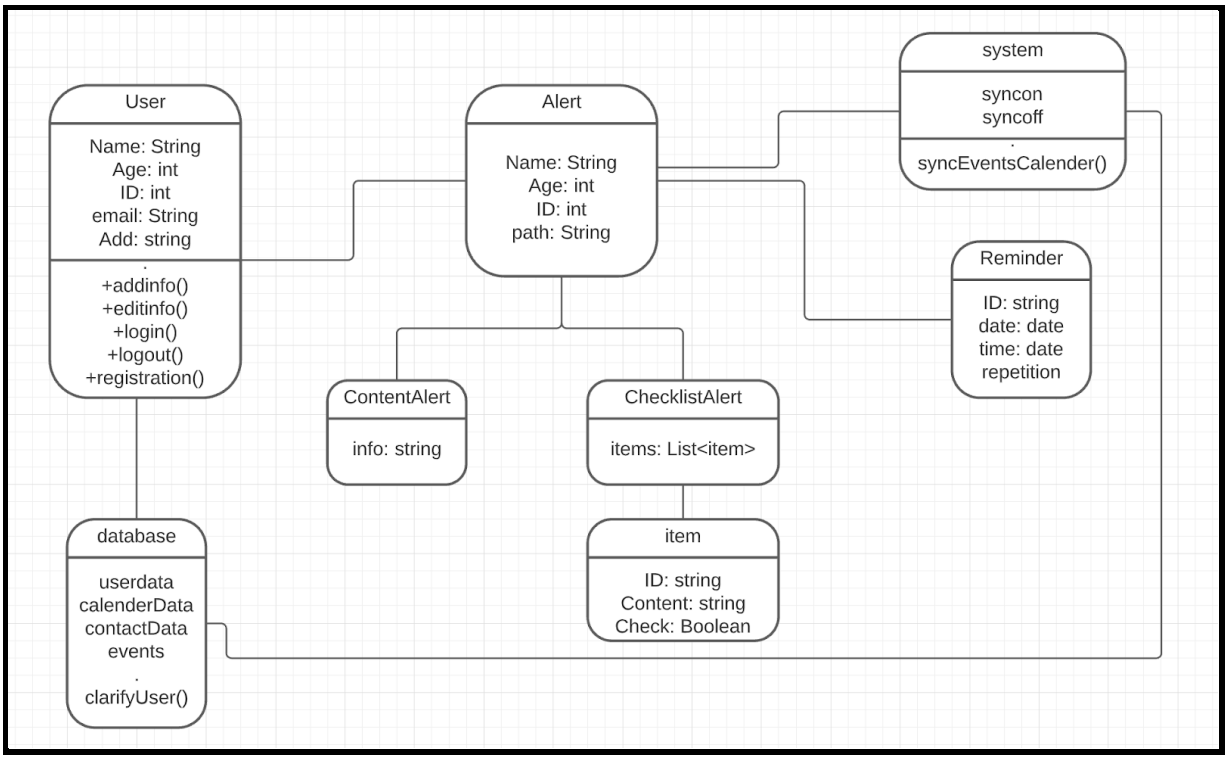
- System
  - Hardware - Android phone or tablet
  - Software - Android 11, Android Studio
  - Database - local
- Security - 2-factor authentication or encryption

3. Project Specification

- Focus / Domain / Area - This is geared toward anyone with an android. For those who are forgetful of the world around them. For those who want to stay close to their loved ones.
- Development Environment - Android Studio, SQLite inbuilt data base.
- Platform - Android Mobile
- Genre (Game, Application, etc) - Application

4. System – Design Perspective

- Identify subsystems – design point of view
  - UML Diagram: https://lucid.app/lucidchart/44fbe809-e39f-4410-b8c6-896e477f2ac4/edit?invitationId=inv_1ddd47d6-9900-47b0-a8f5-4858aa330202



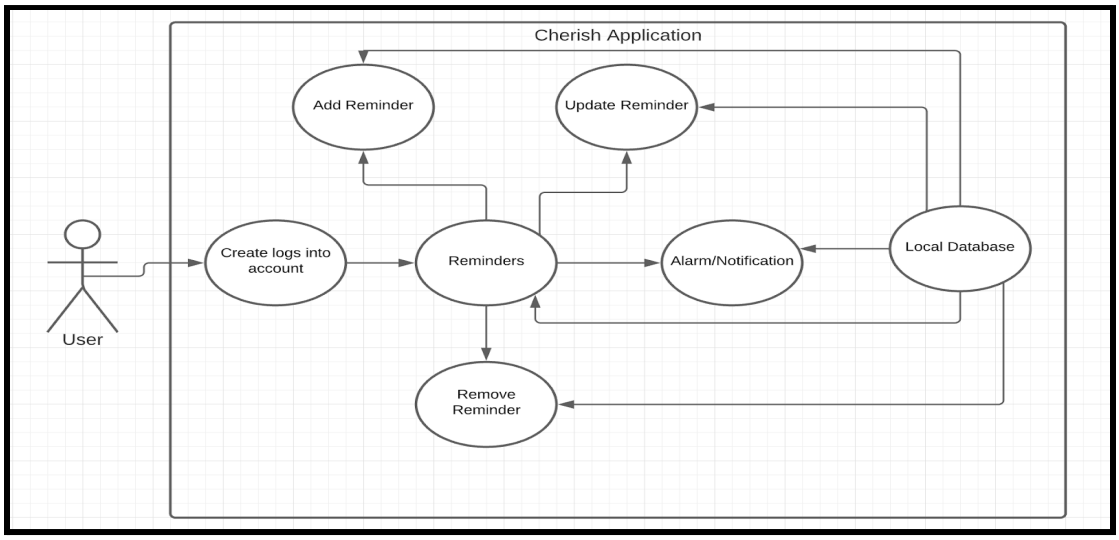
- Sub-System Communication:
    - Controls:
        - Shared Preferences: Saves primitive data as key-value pairs. Requires a key, which is a string and a corresponding value for a said key. The value can be a bool, float, int, long, or string. This data is stored in an XML file on the Android device in a private directory. Apps can have more than one of the Shared Preferences files and they're used to store app preferences.
        - Internal Storage: Very similar to working with any other file system. Provides a way to store data on the file system, but only your app and user can access it. You get references to File objects, and you can store data of almost any type with a FileOutputStream. The only difference to a normal file system is that the contents are only accessible by the app.

- SQLite Databases: Android provides support for apps to use SQLite databases as another option for data storage. These databases are specific to the app and can only be accessed from inside the app.

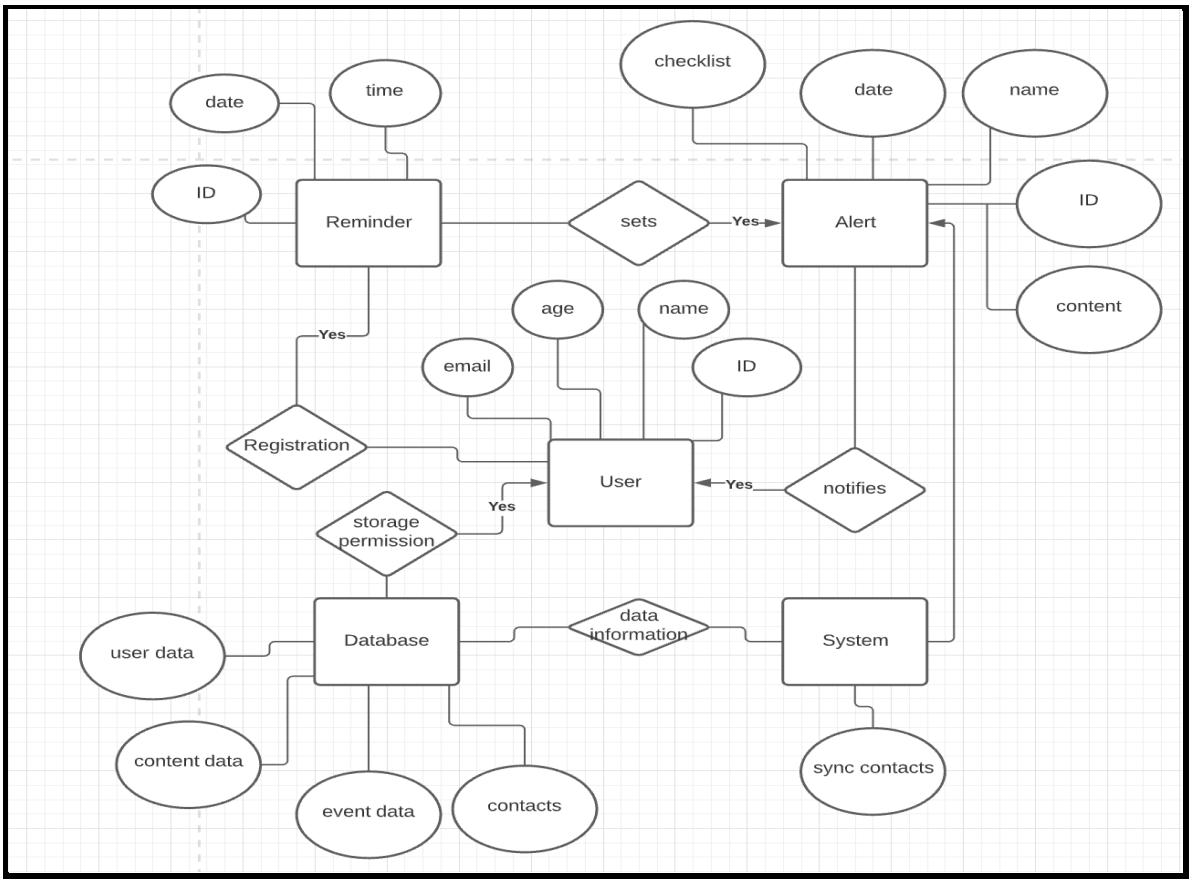
○ I/O:

- Inputs: Login details, new feature requests, new reminders
- Outputs: account details, reminders sent, app notifications, email notifications
- Processes: Sending app notifications, adding and removing reminders, email verification, password reset

○ DataFlow:
https://lucid.app/lucidchart/81d81ead-9ccc-4aa2-976e-8a78e554a5ac/edit?invitationId=inv_35e5b57f-5e11-492f-861e-264533b5abdd



● Entity-Relationship Model (E-R Model):
https://lucid.app/lucidchart/dda6ea3f-e420-4b42-9b05-30a9fc9e5f14/edit?invitationId=inv_ad1a9417-d9f4-4ed9-8cf3-4176ddade168



● Overall operation - System Model

○ Prompts based on a set of defined activities

○ Can handle multiple activities on calendar and events

○ System can run in the background of other applications

○ Can access user, content, event, and contact data.

5. System – Analysis Perspective

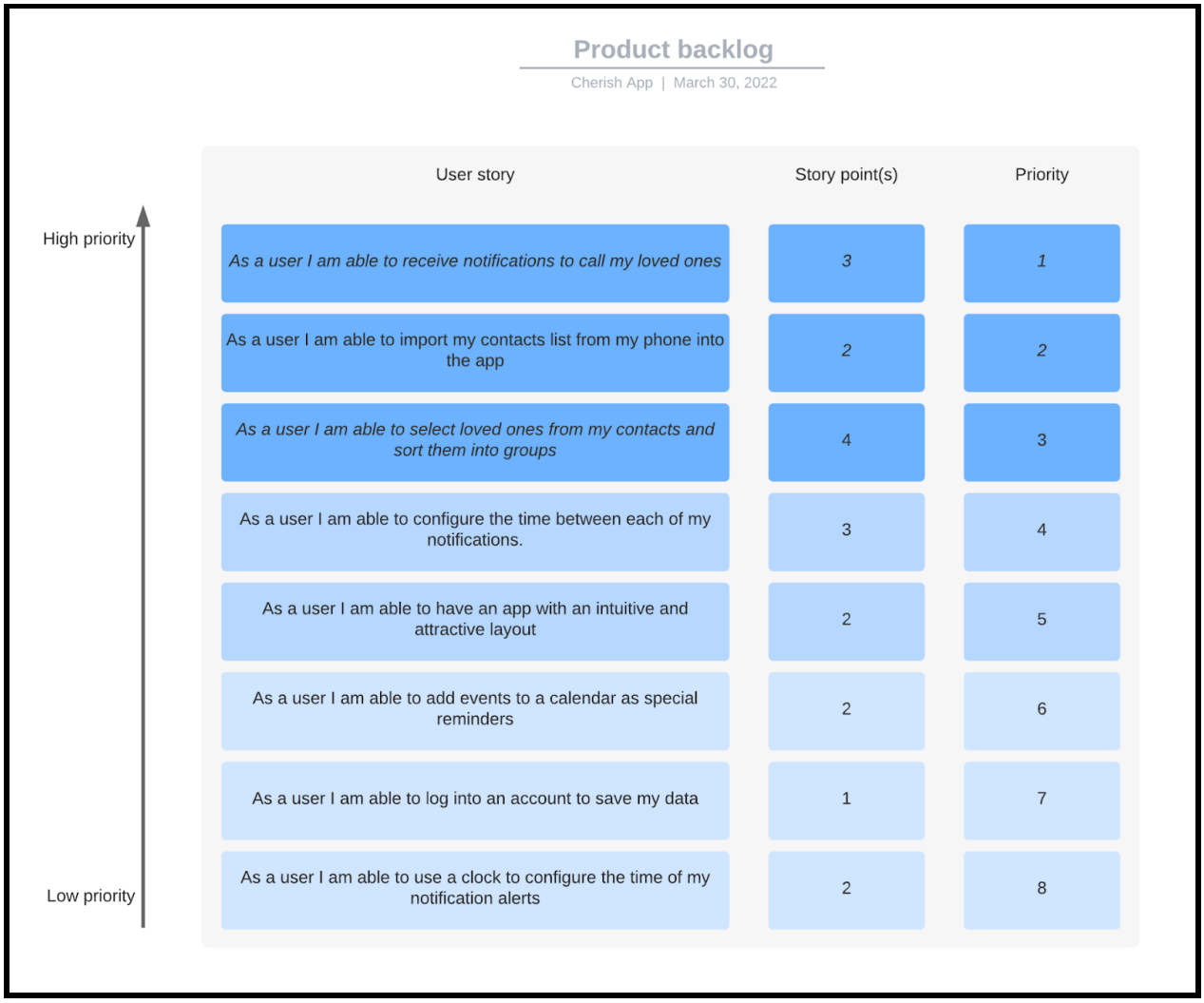
- Identify subsystems – analysis point of view
    - ☐ Notification system
    - ☐ Registration system
    - ☐ Contact system

- System (Tables and Description)
    - Data analysis
        - Data dictionary (Table - Name, Data Type, Description)
            - Reminder
                - ☐ Time=string, Date=string, Name=string
            - Alert
                - ☐ Checklist=string, Date=string, Name=string, ID=int, Content=string
            - User
                - ☐ Email=char, Age=int, Name=string, ID=int
            - Database
                - ☐ User=string, Content=string, Contact=string, Event=char
            - System
                - ☐ Sync Contacts=string
        - Process Model
            - Agile Model
            - Using Trello to assign tasks to each group member.

- Algorithm Analysis
    - Big - O analysis of overall System and Subsystems
        - Overall System Sync Contacts: O(n) time complexity
        - Notification system: O(1) time complexity
        - Registration system: O(1) time complexity

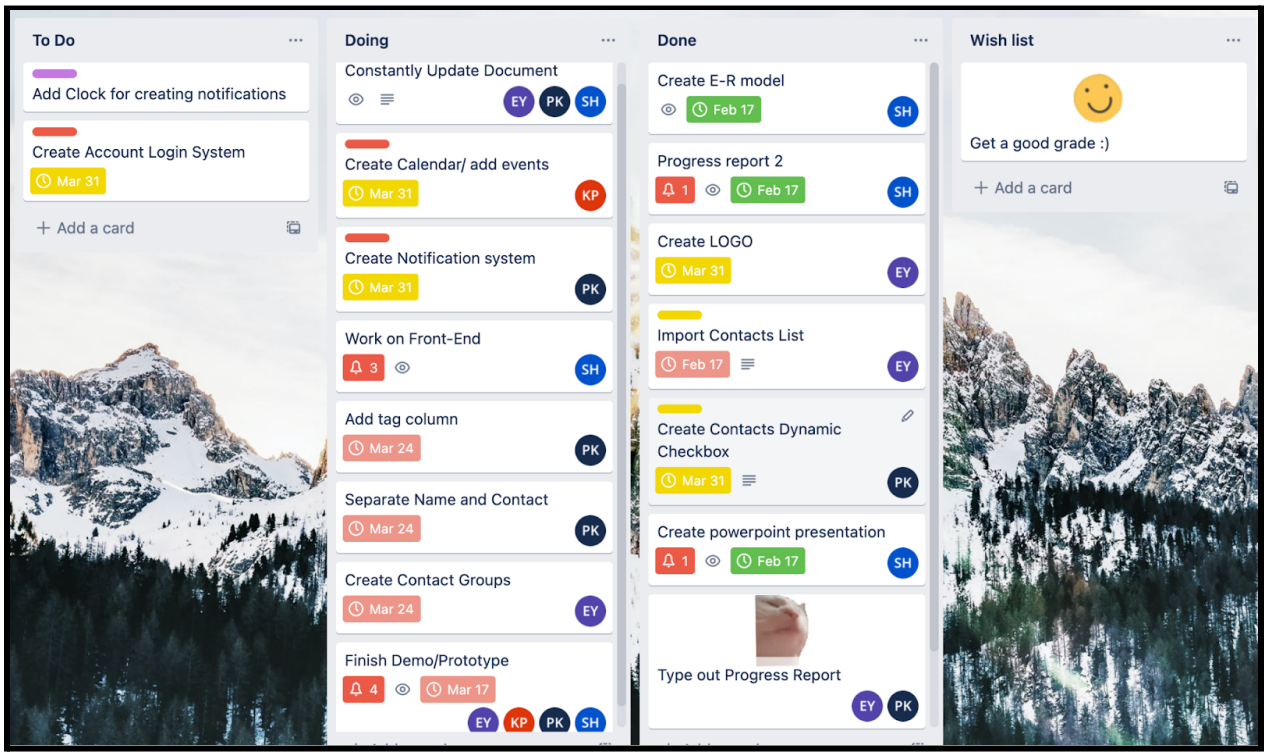
6. Project Scrum Report

●Product Backlog (Table / Diagram)

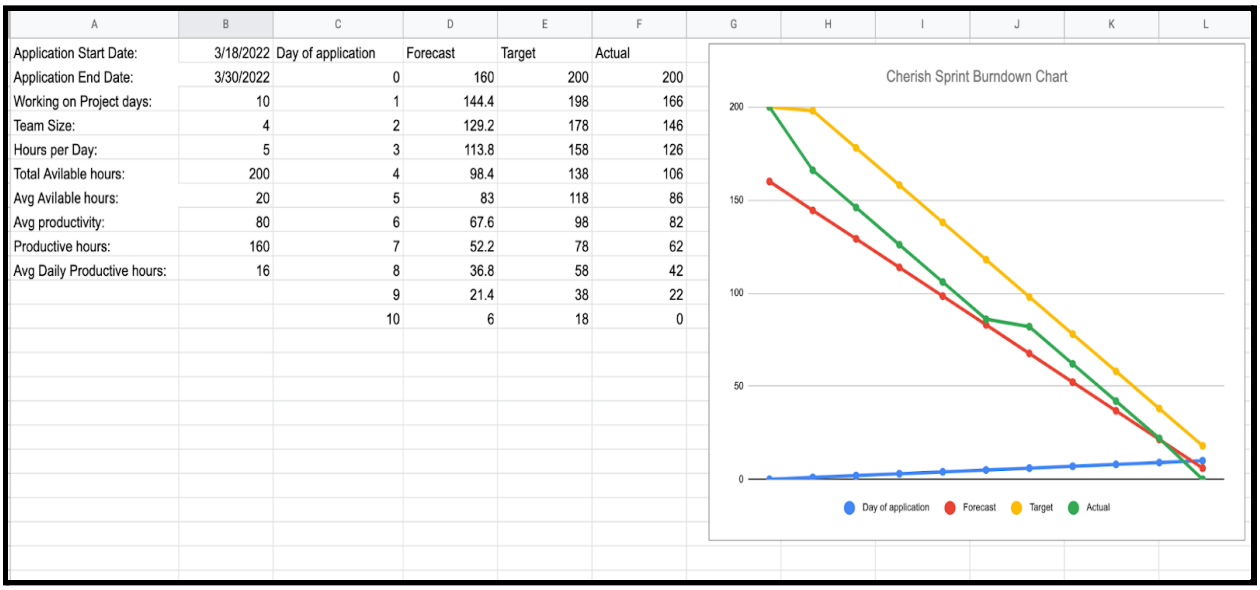


User story	Story point(s)	Priority
As a user I am able to receive notifications to call my loved ones	*3*	*1*
As a user I am able to import my contacts list from my phone into the app	*2*	*2*
As a user I am able to select loved ones from my contacts and sort them into groups	4	3
As a user I am able to configure the time between each of my notifications.	3	4
As a user I am able to have an app with an intuitive and attractive layout	2	5
As a user I am able to add events to a calendar as special reminders	2	6
As a user I am able to log into an account to save my data	1	7
As a user I am able to use a clock to configure the time of my notification alerts	2	8

● Sprint Backlog (Table / Diagram)



● Sprint Burndown Chart: Consists information from 2 weeks by using excel to create:

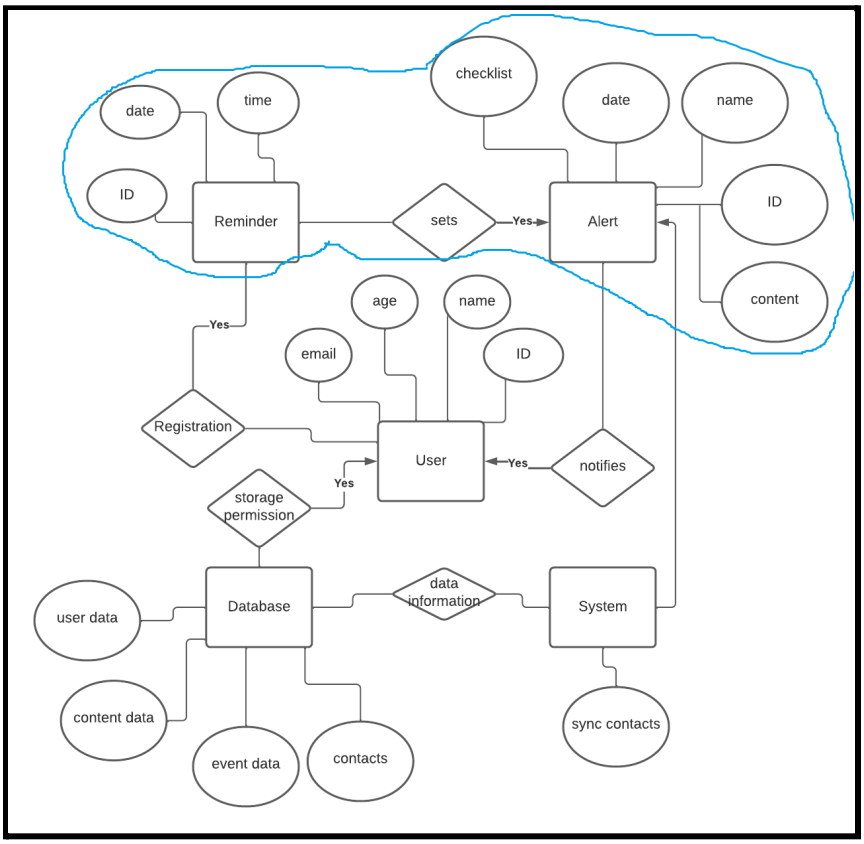


Application Start Date:	3/18/2022	Day of application	Forecast	Target	Actual
Application End Date:	3/30/2022	0	160	200	200
Working on Project days:	10	1	144.4	198	166
Team Size:	4	2	129.2	178	146
Hours per Day:	5	3	113.8	158	126
Total Avilable hours:	200	4	98.4	138	106
Avg Avilable hours:	20	5	83	118	86
Avg productivity:	80	6	67.6	98	82
Productive hours:	160	7	52.2	78	62
Avg Daily Productive hours:	16	8	36.8	58	42
		9	21.4	38	22
		10	6	18	0

7. Subsystems

7.1 Subsystem 1– Notification system

- Initial design and model



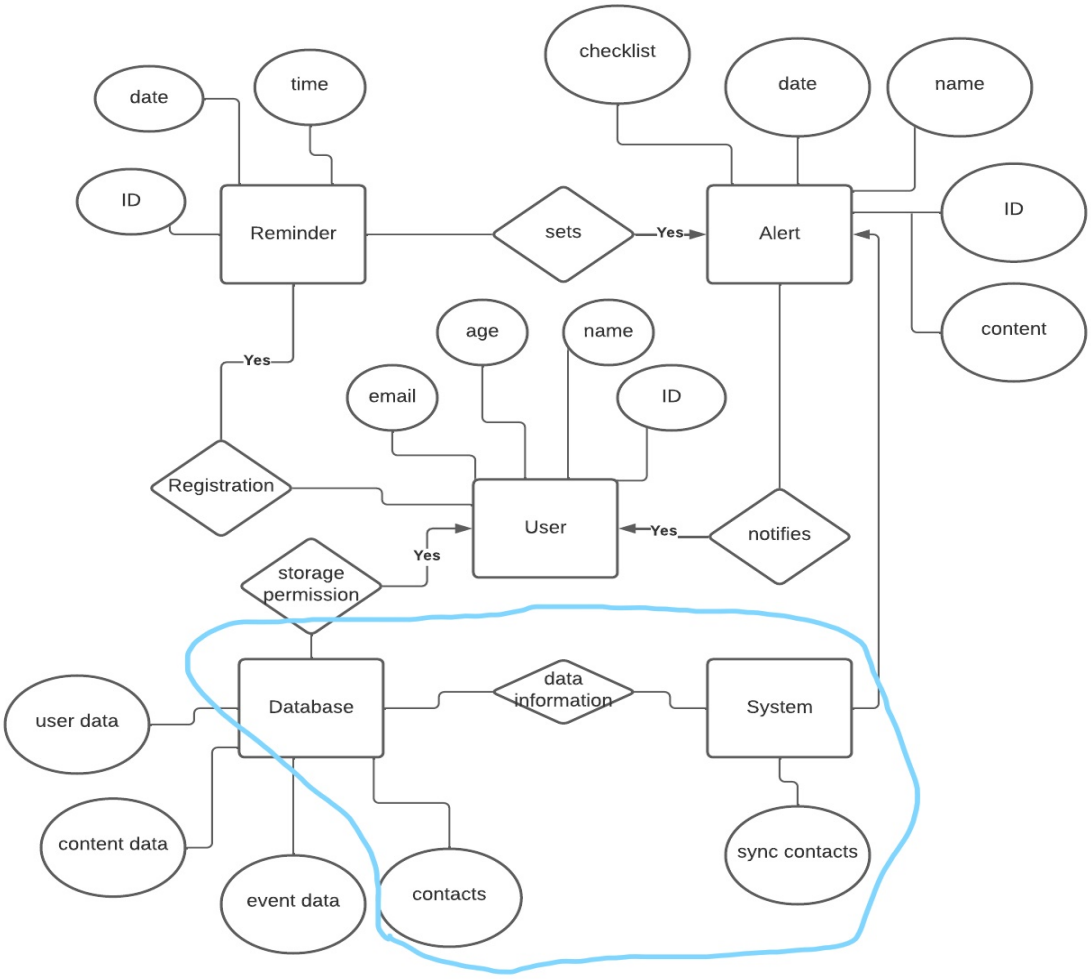
  - Design choices
    - Add new events to remind oneself to text loved ones
- Data dictionary

Data	Type	Description
Name	String	Stores and can access the name of the reminder in an array list of the reminders that are set in the app
Time	String	Stores and can access the time of the reminder that will remind the user
Date	String	Stores and can access the date of the reminder that user wants to be reminded
Month	String	Stores and can access the month and keeping track of the months in a year
Days	String	Stores and access days of the year and make sure the appropriate number of days is correct with each month. Helps show the user the highlighted day the user has selected.

- If refined (changed throughout the project)
    - Reason for refinement (Pro versus Con)
        - The pros of changing the ID to Name for organizational reasons to help coordinate the name or title to a certain event. The content was redundant and can be stored to Name, Time, and Date. The checklist would be stored in the name of the events to keep track of all the events
    - Changes from initial model
        - Checklist, Content, and ID to Week, Month, and Name
    - Refined model analysis
    - Refined design (Diagram and Description)
- Scrum Backlog (Product and Sprint - Link to Section 6)
    - As shown in section 6 the sprint was met with a few upgrades to be made.
- Coding
    - Approach (Functional, OOP)
        - OOP
    - Language
        - Java
- User training
    - Training / User manual (needed for final report)
        - When accessing the calendar, clicking the arrows on the top left and right would take you to your respective months. Right being next month and left being the previous month. Clicking on week would minimize the view to only seeing each week in the month. From there you can click the arrows on the top left and right or make a reminder using a name, time, and date. The name is the title of the reminder, the time will be when you want the app to remind you and the date is the day that the reminder happens.
- Testing
    - Using Android Studio to create an OS emulator to run the app and view the calendar and reminder system and test the functionality.

7.2 Subsystem 2 – Contact System

- Initial design and model



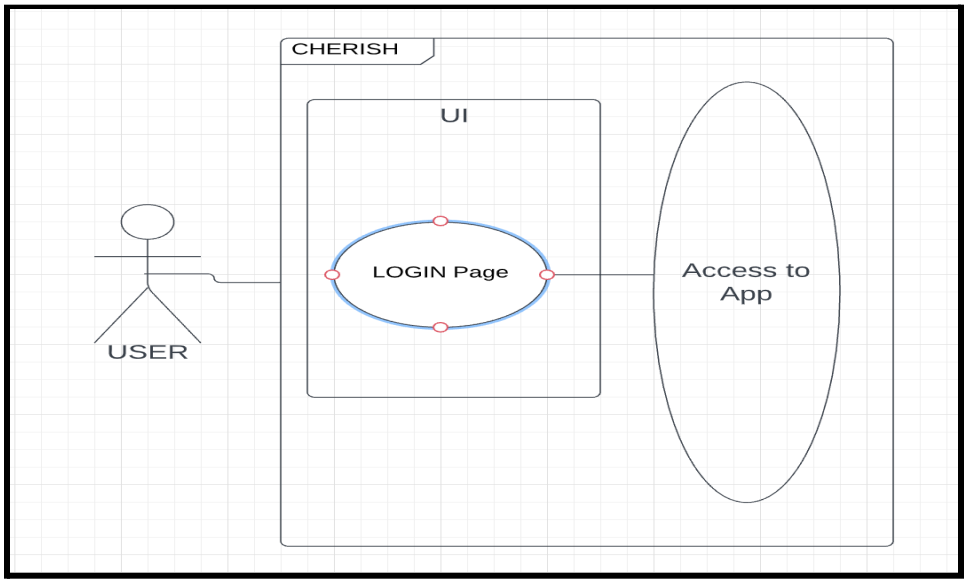
  - Design choices
    - Import contacts from the phone into the array list.
- Data Dictionary

Data	Type	Description
Contact Name	String	The name of the contact
Contact Number	String	The number of the contact

- If refined (changed throughout the project)
    - Reason for refinement (Pro versus Con)
        - Saving to a database instead of a local ArrayList is better for storage since the data can be accessed even after the app is closed.
    - Changes from the initial model
        - Save contact information to SQLite database, instead of local ArrayList.
    - Refined model analysis
    - Refined design (Diagram and Description)
        - Stores contact information as a string in the SQLite database
        - Displays the contact information in a ListView
        - List of contacts can be updated and deleted.
        - Individual contacts have specific notification settings
- Coding
    - Approach (Functional, OOP)
        - Object-Oriented programming.
        - Agile methodology
    - Language
        - Java
        - Android 10
- User training
    - When given the prompt to allow the app to access your contacts. Allow it. Once you've done that you will be brought to the login page. Follow instructions in 7.3 for the login page. Once done, you can then select which contacts to save to the app.
- Testing
    - Performed JUnit tests on inputs for changing contact information and used the android emulator to test contact synchronization and notifications.

7.3 Subsystem 3– Registration System

- Initial design and model: Able to Create a UI Login Screen.
    - Use case/UML Diagram:



○ Design choices: Simple UI that consists of

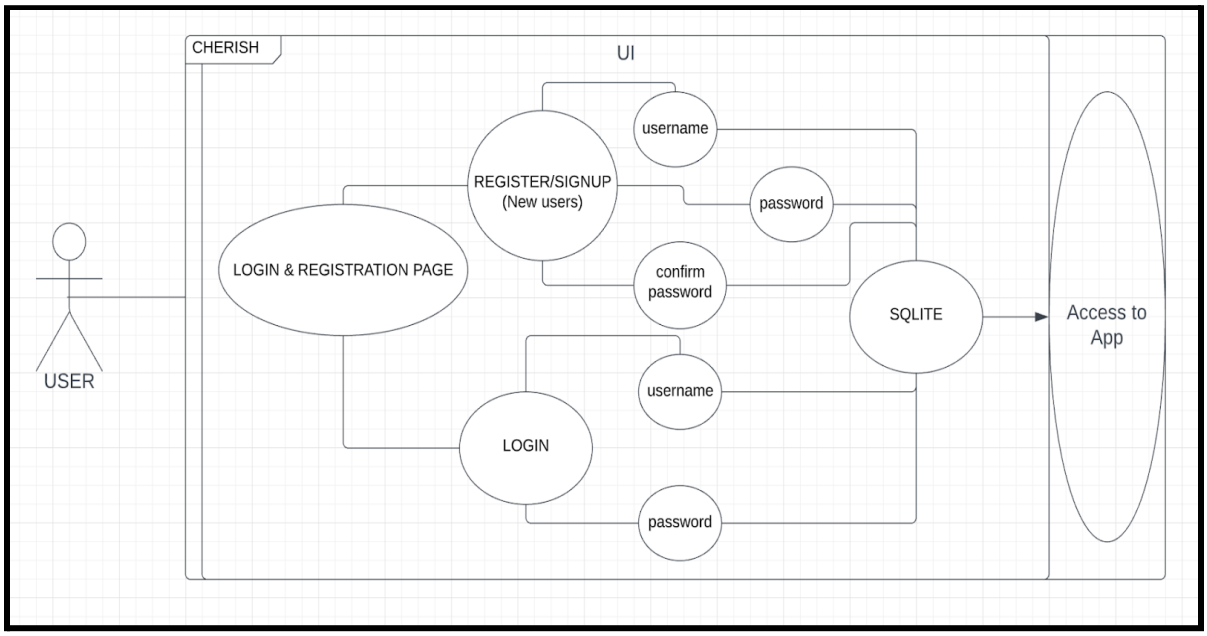
- Application title: Cherish
- Application header showcasing the user is on a Login and Registration page
- Area to enter Credential: Username and Password
- Buttons for login and registration

●Data dictionary:

Data	Type	Description
Username	String	Stores and can access Username information of the user
Password	String	Stores and can access Password information of the user in the format of the string.

- If refined (changed throughout the project):
    - Reason for refinement (Pro versus Con):
        - Advantages:
            - Can store login and register new users' information such as username and password.
            - Uses SQLiteOpenHelper to register and access user information.
            - Added background and logo for UI preference and can change Dark/Light mode.
    - Changes from the initial model:
        - More user-friendly
        - Better User information security
        - Better UI appearance
        - If a user forgets a password they will be able to change it on the profile
    - Refined model analysis:
        - By creating a simplistic UI for the user, they can now create and store their credentials to access the application.
        - The password is hashed out so others may not view it.
        - Passwords can be changed if forgotten.

- Refined design (Diagram and Description):



- Scrum Backlog (Product and Sprint - View Section 6):
- Coding
  - Approach (Functional, OOP)
    - OOP - JAVA
    - Agile methodology
  - Language
    - JAVA
    - Android 10
- User training
  - Training / User manual (needed for final report)
    - Due to simplicity of the UI the user task for registration and login:
      - New User:

        - Must first click on the Signup button to register a username set a password, email address, and phone number
        - Then the user must click the register button to store the information into the application database.
        - Lastly, the user needs to click on the login button to go back to the login page and put in their new credentials to access the application.
    - User:
        - If a user has already registered they can log in to the application using their credentials: Username and Password.
    - Forgotten Password:
        - If a user has forgotten a password they need to click on the “Forgot Password” button to register their newly changed password.
        - Then they must log in by clicking on the login and enter the edited credentials to access the application.
- Testing
    - Using Android Studio to create an OS emulator to run the Application and view the prototype
    - Once the whole project is completed we will test the final product on an Android Phone.

8. Complete System:

- Final software/hardware product:

  Final software: https://youtu.be/XQPVxfDmCIE [OS Emulator video. Can download the application and will run exactly as shown in the video on an Android device.]

  Android application: Instructor has been added to the file for easy download on an Android mobile device. https://drive.google.com/file/d/1wJLLhbFHzqkKQmlYKtNHyw8wKN9xahs-/view?usp=sharing

- Source code
  - GitHub Link: https://github.com/Srushti1013/cherishProject
- User Manual:

1.1 INTRODUCTION:

What is the *Cherish* mobile app?

Cherish is an Android mobile application that allows users to keep connections with friends and loved ones close together during difficult times online by sending quick reminders to the people the users select in their local contacts on specific days. The reminders can be set up daily, monthly, and yearly by using the inbuilt calendar.

Which devices does the mobile app support?

You can install Cherish on any mobile device that has an Android Operating System.

How it works

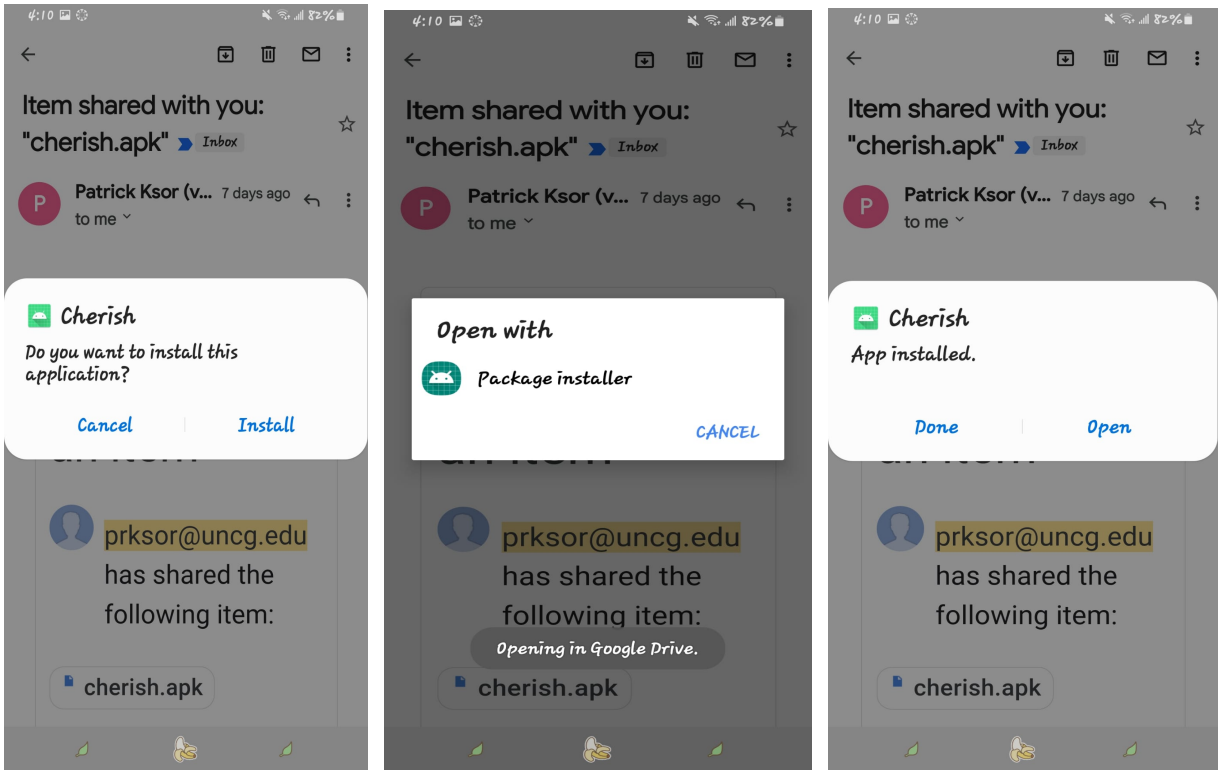
Cherish requests access to the user's local contacts. Once the request is granted you can register an account to store users' information which can be edited in the profile section of the application once logged in.

Cherish Application allows users to

- Register and log in to the Application
- View contact data from a local android device and add required numbers to the contact list.
- Set up calendar events for users' specified contacts to remind them to connect with person
- Can add a description to created event
- Notifications will be sent at the user-specified date on the calendar event
- Edit contact information from the contact list
- Edit user information on Profile page

1.2 INSTALLATION:

The user can access our repository on GitHub using the link: https://github.com/Srushti1013/cherishProject and then can run the program on Android Studio. If a user does not have an android device they can run the application on Android Studio using the emulator feature. If a user has an Android device they can extract the application as an APK file onto Google Drive and then download onto the android device and run it. The application will ask to access contact information and this is necessary for the application to function. If using the emulator feature, please add a few contacts in the emulator to see how the application does run. Users can also contact team members through the email listed on the GitHub repository under the ReadMe file.



1.3 SECURITY

To access the backup data, sign in to your account by providing the e-mail address and password for that account. Furthermore, data is in encrypted form. Only you have access to your encrypted data.

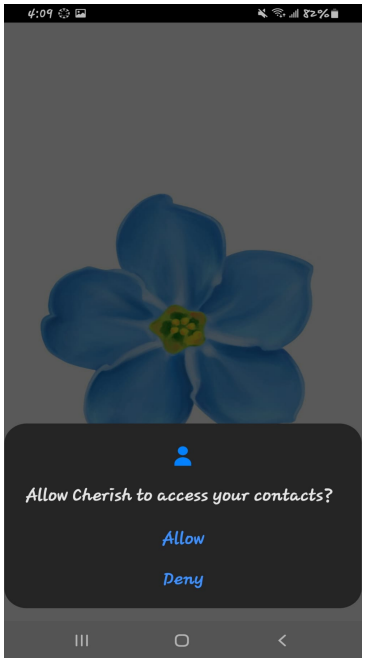
2. PROGRAM WORKSPACE

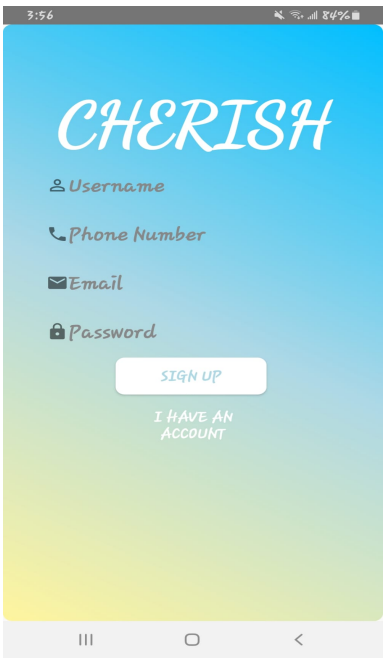
2.1 REGISTRATION

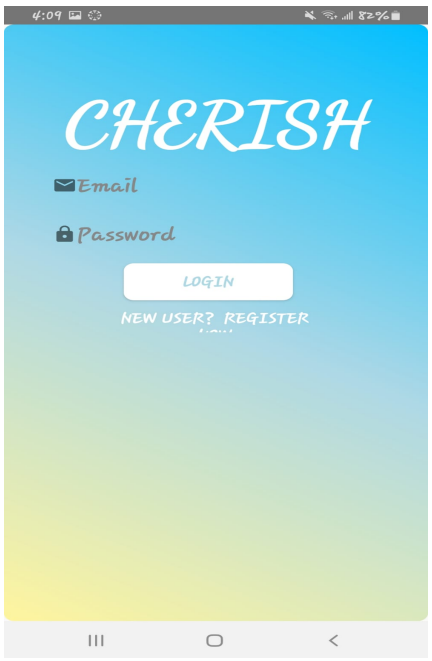
To use the application the user must first allow access to contacts and then enter user credentials such as username, phone number, email, and password. Then click the sign-up button to go to the login page and reenter your email and password to make sure it is you.

If you are a current user, just log in using your email address and password.

By default, you need to sign up only when you start the application for the first time.

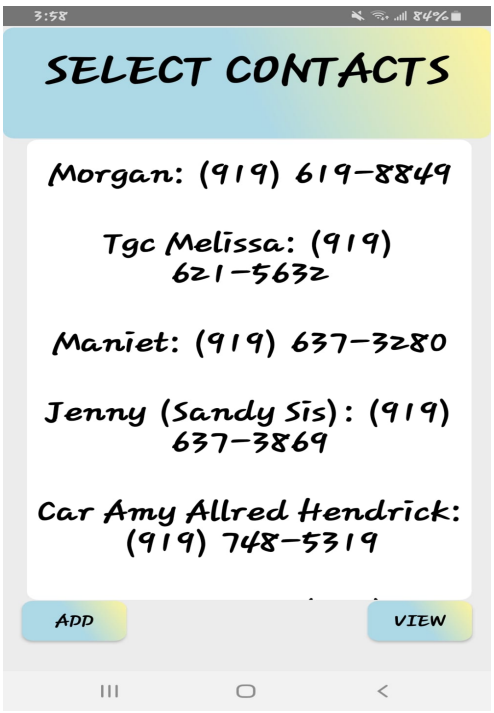






2.2 APP OVERVIEW

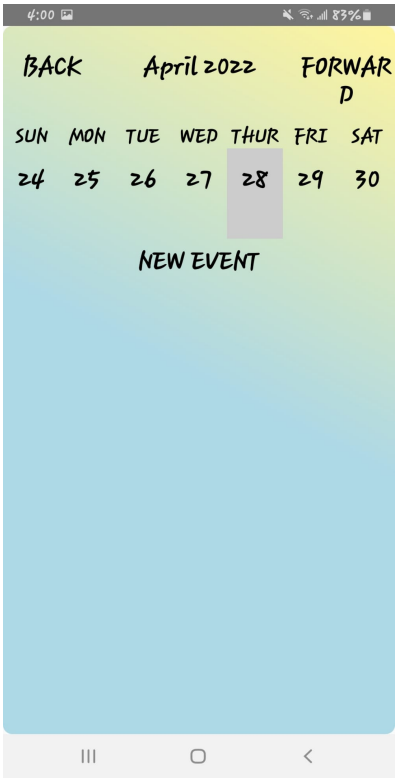
Contacts page: Lists all of the users' contacts from their mobile device.

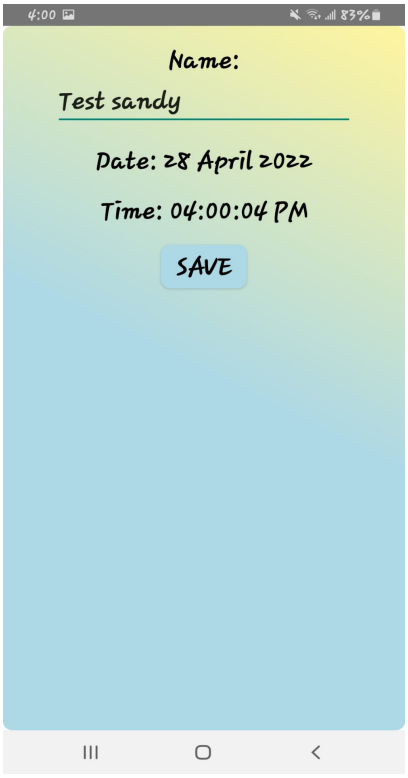


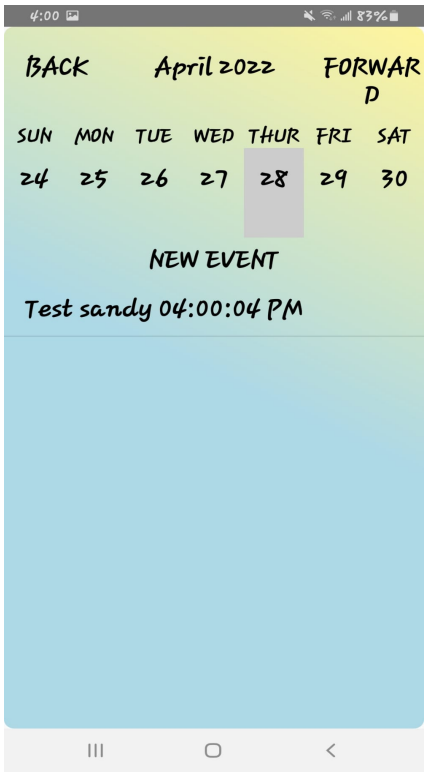
Saved Contacts: These are contacts that are selected and add button is selected to store them on the saved contact page. The contacts on this page can be deleted or calendar notifications can be set up. This page also has a profile button that contains user information.

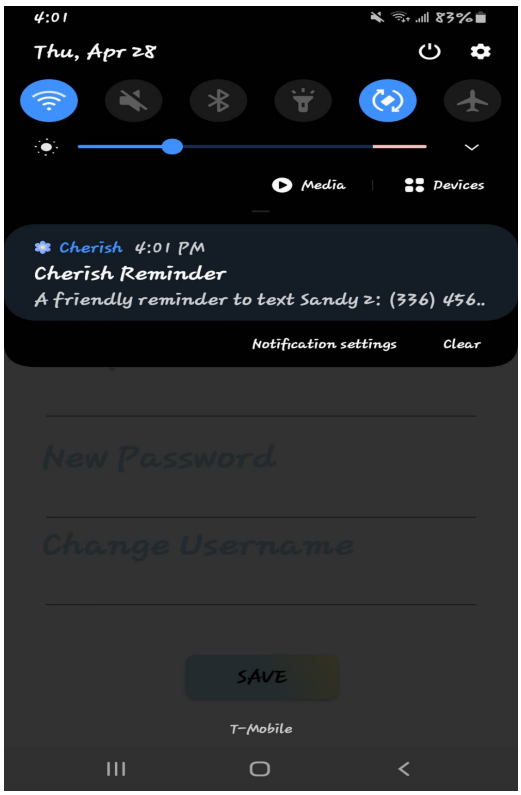
Calendar Page: In the calendar page the user can view the contact profile and edit changes such as change name or number. The user can also set reminder notifications by clicking notification button and creating a new event as shown below. Once set, the user will receive notifications that will repeat as requested. Few images are shown below of the calender page.


4:01
83%
USER PROFILE
Sandy z
(336) 456-5286
DELETE
NOTIFICATIONS
SAVE

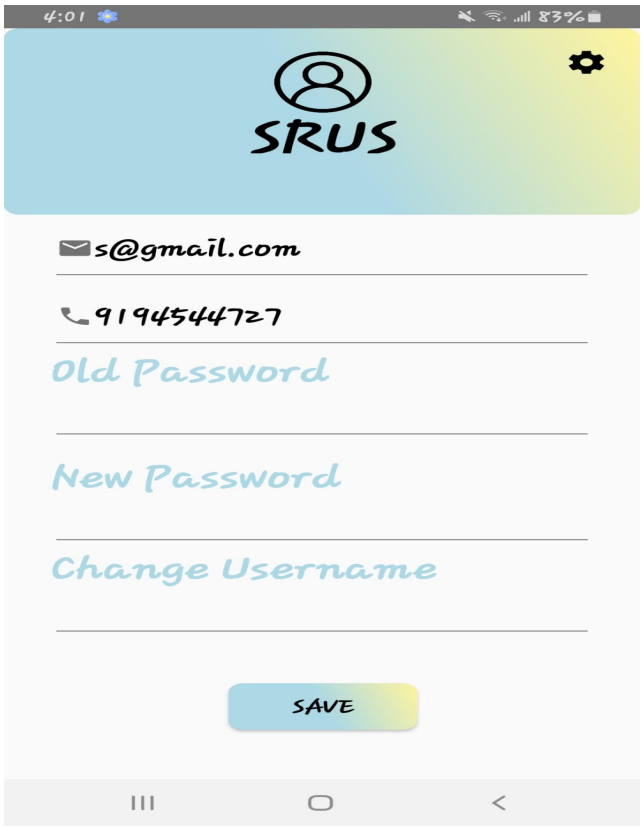

4:00
83%
BACK
April 2022
FORWARD
SUN MON TUE WED THUR FRI SAT
24 25 26 27 28 29 30
NEW EVENT


4:00
83%
Name:
Test sandy
Date: 28 April 2022
Time: 04:00:04 PM
SAVE


4:00
83%
BACK
April 2022
FORWARD
SUN MON TUE WED THUR FRI SAT
24 25 26 27 28 29 30
NEW EVENT
Test sandy 04:00:04 PM


4:01
83%
Thu, Apr 28
Media
Devices
Cherish 4:01 PM
Cherish Reminder
A friendly reminder to text Sandy z: (336) 456..
Notification settings
Clear
New Password
Change Username
SAVE
T-Mobile

Profile Page: On this page, the user can edit their information and change their user name and password. In the below images, I changed the username:



- Evaluation by client and instructor:

- Team Member Descriptions

Ke’Shawn (Notification System): Worked on implementing the events and calendar within the reminder app. Gives you the ability to name your reminder to text or call anyone in particular within your contacts.

Srushti(Registration System): Worked on the front end by creating a clear User Interface that lets users create and log in to their accounts. By using the email as a primary key, the user can change their username and password in the profile section.

Eric(Contact System): Worked on implementing the contacts to have permission to acquire the contacts from the user’s phone. Worked on the ability to save the contacts to use for reminders.

Patrick(Contact System, and Profile): Worked on applying the functionality of being able to create an account to apply your contact information. Uses contact information to set up different reminders within your contacts.
