Software Requirements Specification

for

Dorm.ly

Version 1.0 approved

Prepared by Joshua Soriano, Ameen Mulla, John Tran, Lester Dan Goze

Group 4, Section 003

February 12, 2021

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Dorm.ly	1/25/2021	School campus, dorms, housing	Version 1.0.0
Dorm.ly	7/25/2021	Land/Tenants (detached homes, duplex)	Version 1.1.2
Dorm.ly	1/25/2022	Small businesses, apartment building rental units	Version 1.2.0

1. Introduction

1.1 Purpose

- This new software product will be a social media or a group chat application for people living in a shared space or a dorm. This is version 1.0.0. The app is called **Dorm.ly.**

- This new software product will allow users to make status posts on errands that need to be done, learn about current events, or be a place to chat with each other.

- This new software product will allow users to DM other users privately for confidentiality.

- This new software product will be providing a WebApp and a MobileApp. Users will be able to call each other if they want to meet, they can also video chat if they wish to talk in person rather than text.

- This new software product will allow landlords and roommates to rate each other which can be viewed on their profile. This will allow future applicants to view their future potential landlord/ roommate.

- Our application differs from other apps like Facebook because it will be more locally based and the community will be closer to each other, meaning it is easier to connect and meet people in your building.

1.2 Document Conventions

Acronyms	Description
SRS	Software Requirements Specification
DM	Direct Message
DDoS	Distributed Denial of Service
OS	Operating System
APP	Application

IT	Information Technology
FR	Functional Requirement
NFR	Non-Functional Requirement
FAQ	Frequently asked questions
NFC	Near Field Communication
C#	High level programming language
RAM	Random access memory
Wi-fi	Wireless fidelity
USB	Universal Serial Bus
HTTP	Hypertext transfer protocol
FTP	File transfer protocol
VPN	Virtual private network
IP	Internet protocol

1.3 Intended Audience and Reading Suggestions

- This document is intended for developers, project managers, testers, and documentation writers.

- This document is intended for stakeholders and users that will be using the application for their own dorm/living arrangement.

1.4 Product Scope

- This is a social media application that will help with managing errands in a shared living space (ex. School dorms, boarding homes, family homes, etc.)

- The intent is to assist with day-to-day tasks such as: groceries, laundry, mowing lawn, shoveling snow, and other housekeeping chores.

- The application will allow household members to divide up tasks that need to be completed. In the event there is poor communication between household members, or language barriers the application will clearly describe what tasks need to be handled. The application will also include multiple language translation.

- In future iterations, the application can also be managed by landlords/property managers to ensure that their property is looked after by their employees, or maybe the tenants themselves - if it was a part of their lease agreement.

1.5 References

- Marketplace - www.facebook.com
- Listings Methods - www.Kijiji.ca
- Competitor - www.postmates.com
- Competitor - www.doordash.com
- Competitor - www.skipthedishes.com
- Competitor - www.skipthedishes.com
- Fund Transactions - www.interac.ca
- Maps - https://maps.google.com
- Translator - https://translate.google.ca
- Translation apps - https://uncubed.com/daily/best-translation-apps-for-travel-in-2019/

2. Overall Description

2.1 Product Perspective

This is a new, self-contained product designed to solely function as a task manager specifically for unrelated users. This app will reduce physical interaction between residents and landlords to abide with provincial restrictions; communication and completion of errands/tasks can all be done through this application without physical interaction. It's geared towards keeping the user's safety and well-being in mind during the pandemic.

2.2 Product Functions

The new software will have features such as:

1. User status updates - provides the user ability to post pending errands or requests from other users.

2. Message Board/ Chatroom - allows users to leave posts in response to specific errand within or outside of their network.

3. Video calling - allows users to contact other users through their preferred devices without engaging in physical contact.

4. Rating system - allows users to rate each other based on quality of work.

5. Translator - the users will have the option to choose their preferred language to use the app. The system will use translate and display incoming messages to the user's requested language.

6. Alert/notification system - give users a message or notification on their mobile device when they get DM's or tasks/requests from other users.

7. Admin accounts - landlords or dorm supervisors will have special accounts with more control over the app.

8. User profile - users will have their own custom profile to display their talents and capabilities.

2.3 User Classes and Characteristics

- **Customers** - users will be students, landlords, residence hall, business owners, building management
  - Students away from home and are living in dormitories.
  - Young adults renting shared living space.
  - Landlords that assign certain housekeeping responsibilities to the tenant.
  - Landlords who want trusted reviews about their applicants (tenants).
  - Tenants who want trusted reviews about their landlords or roommates.
  - Single and Multi-family home.
- **IT department** - the IT department will manage the network services and cybersecurity of the application.
- **Marketing department** - the marketing team will advertise the application and attempt to get users to try the product.
- **Development team** - the development team will update the app and update it to remove bugs and implement requested features.
- **Investors/ Sponsors** - will have the ability to manage advertisements based on user searches and most frequently requested tasks.
- **Customer support department** - customer support will assist users as they come across problems with the application or if they have questions.

2.4 Operating Environment

Devices:

- Tablets
- Smart watches
- PC Desktops and Laptops
- Smart Phones

Operating Systems:

- iOS 8.0 +
- iPadOS 13 +
- watchOS 5 +
- macOS X Mountain Lion +
- Android 7.0 +
- Windows 8.1 +

Hardware:

- Apple A8 Chip + / Qualcomm Snapdragon + / Intel i3 2300/ AMD Ryzen 5 1200
- 1GB of RAM +
- 500MB of Storage
- Ethernet or Wi-Fi connection
- Mobile data

2.5 Design and Implementation Constraints

- Software
  - Android and iOS compatibility.
  - Languages needed (tutorial).
  - Out of date operating systems may lead to compatibility or security issues if the operating system is no longer being updated.
  - The user interface may run into compatibility issues depending on the platform such as Windows, Mac, or mobile devices.
- Hardware
  - Insufficient memory can lead to the app crashing
  - Device display resolution to view the layout properly; 1080p minimum.
- Corporate/ Regulatory Policies
  - E-transfer - privacy access to user's banks, maximum transaction amounts.
  - Privacy - User Location, password, identity.
  - Liability - Care of other user's property. e.g., lawn mowing/ dog sitting.
  - Errands/ Tasks available -
    - Limit to age restricted products, e.g., alcohol, cigarettes, cannabis
    - Time of day restriction to available tasks. The app will have strict criteria to ensure safety for all members and to abide with noise restrictions. E.g., mowing lawn not available during the evening/ after 8pm.

2.6 User Documentation

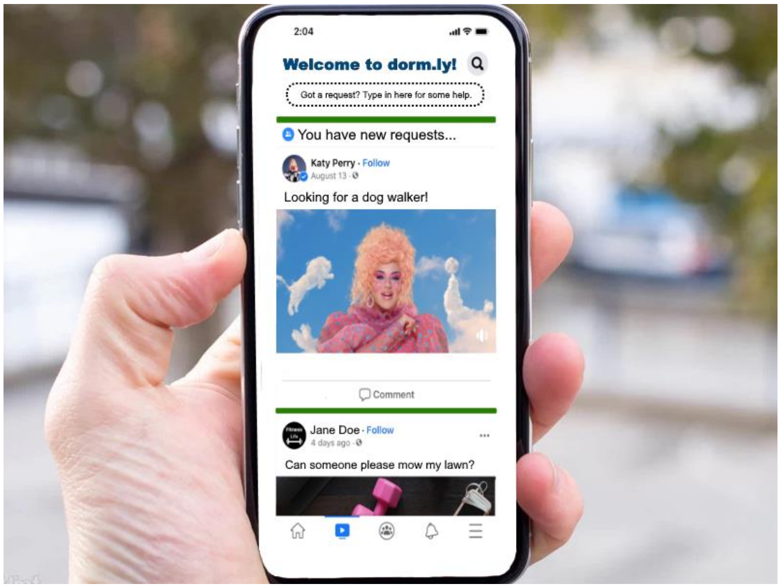
Upon a fresh install of the application, users will be greeted with a mini tutorial on how to use the app. They will be taught the basics of the app and how to get started. For further instructions or guidance, a F.A.Q will be provided alongside a page for customer support and a phone number they can call.

2.7 Assumptions and Dependencies

- Compatibility issues between group of users if device age gap too wide.
- Some errands/tasks are unique to certain households - unable to fully provide support/functionality.
- Instability or compatibility of interaction when the operating system is different between device firmware upgrades.
- Third party app required to support multiple languages to minimize language barriers between users.
- Rating systems will require users to have a minimum 3-star baseline out of 5 in order to have their rating be considered credible and applicable to the user they are rating.
- Access to financial institutions to deposit or request funds (e.g., user buy groceries for household or buys cleaning/maintenance products and requires reimbursement).

3. External Interface Requirements

3.1 User Interfaces



Welcome to dorm.ly!

Got a request? Type in here for some help.

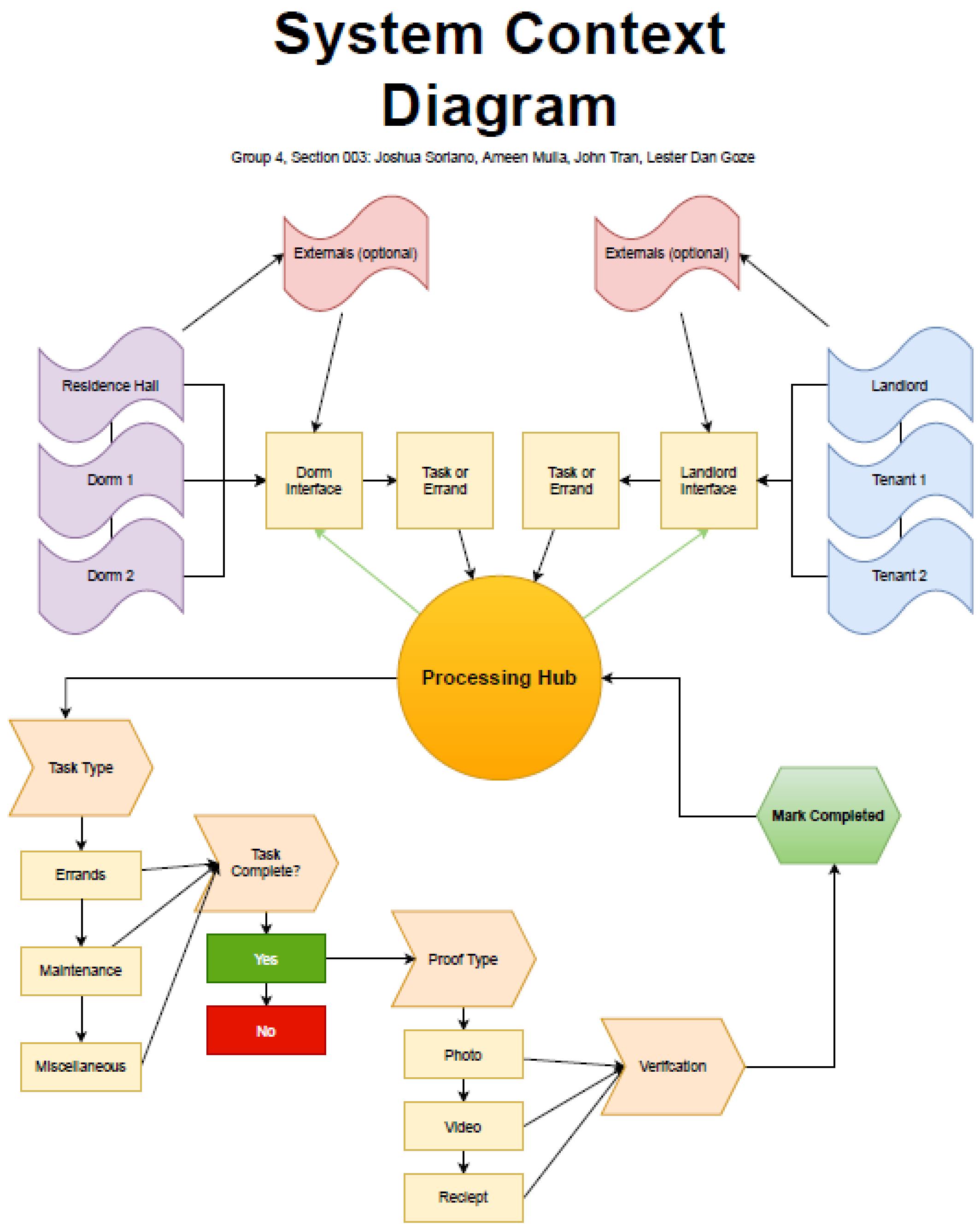
You have new requests...

Katy Perry · Follow
August 13

Looking for a dog walker!

Jane Doe · Follow
4 days ago

Can someone please mow my lawn?



Login Screen

- Username & password login (create new, sign up)
- Log in through Third Party (Google, Facebook, Apple ID)
- Forgot password & username function
- Remember login
- Face recognition
- Fingerprint ID

Menu Button

- Home Screen
- E-Transfer Screen
- Request Screen
- Inbox Screen
- Search Result Screen
- Log out

Home Screen

- User Profile
- Status Update
- Chat
- E-transfer
- Search bar
- Help/ FAQ
- Language Options

E-Transfer Screen

- Choose bank (e.g., Moneris)

Request Screen

- Text box for user to input request
- View Requests; pending or completed
- View available requests within or outside of network

Back-End

- C#
- JavaScript

3.2 Hardware Interfaces

- **Micro USB/ USB C/ USB A / Lightning Cable to USB 2.0+** - allows user to update firmware via connection to pc
- **Bluetooth (NFC pairing)** - connect wirelessly via Bluetooth to headphones or earpieces
- **Camera** - different ways to access information from the application
  - Barcode/QR code reader (quickly access requests)
  - Facial scanner (quicker way to login to your account)
  - Fingerprint scanner (quicker way to login to your account)
  - Proof of completion (swipe a button on the app and take a picture of the finished task)

3.3 Software Interfaces

- Voice-command (ex: Siri, Cortana, Alexa, Google)
  - Provide more accessibility within the app for people who are handicapped
  - Easy navigation (ex: ask Google to tell you what your current requests are)

- Language translator (iTranslate Voice 3, Google Translate, Text Grabber, Microsoft Translator)
    - The app will be translated into different languages
- Payment Methods

  -Financial institution gateways (TD Canada Trust, RBC, Scotia Bank)
  -PayPal
  -Crypto-Currencies

3.4 Communications Interfaces

- Communication functions
    - Email (for account authorization, Monthly billing statement)
    - Web browser (safari for iOS, Android web browser, Google Chrome)
    - Network server communications protocols
        - Cannot use VPN to spoof your IP address

Message formatting

- Message templates (verification emails, no-reply emails, contact number)
- Communication standards
    - FTP (to share files such as pictures)
    - HTTP (to access webpages using the built-in search engine)

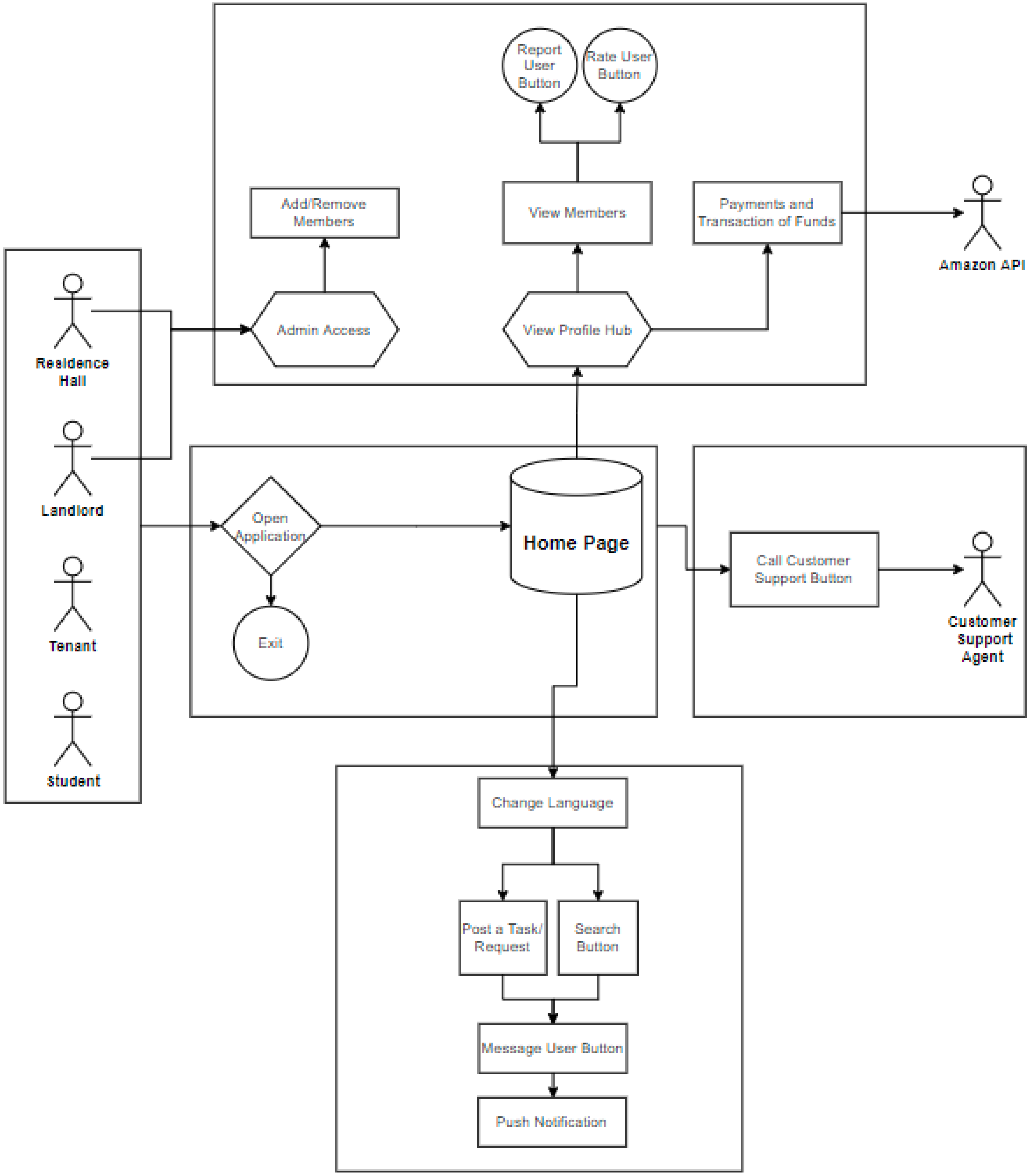
4. System Features

Functional Requirements list				
Requirements ID	Requirement title	Short Description	Priority	Requester
FR01	Rating system	Users should be able to rate each other for quality of work (ex: doing the requested task properly).	High	Landlords (Richard McCreggor) Tenants Residence Hall
FR02	Alerts	The app should give notifications to users when they get requests or messages.	High	Landlord (Richard McCreggor) Tenants
FR03	Admin accounts	Landlords or chief of residences should be given special accounts that have more control over the application.	High	Investor (Zayden Donald)

FR04	Messaging system	Users need to be able to message each other privately.	High	Junior Programmer (John Tran)
FR05	User profiles	Users need profile to display their talents and capabilities.	High	Junior Programmer (John Tran)
FR06	Live customer support/call center	The application should have a section for users to contact or a phone number to dial for support.	High	IT Manager (Derek Regan)
FR07	Flexibility	The application needs to be able to adjust features to different living conditions (ex: a school campus or an apartment complex).	High	Senior Programmer (Joshua Soriano)
FR08	Message board and chatroom	The app needs a place for people to chat and post requests in real-time so communication is easy.	High	IT Manager (Derek Regan)
FR09	Funds Transaction	The application allows connection with financial institutions to electronically transfer funds between users.	High	Landlords (Richard McCreggor) Tenants Residence Hal
FR10	Language Translator	The application will translate all text messages to the appropriate language desired.	High	Senior Programmer (Joshua Soriano)
FR11	Search Function	Allows users to search for keywords related to a task, giving results to qualified top rated members, which then users can select from.	Low	Programming lead (Ameen Mulla)

Dorm.ly Use Case

Joshua Soriano, Ameen Mulla, John Tran, Lester Dan Goze



Use Case Template for Posting

Use case: Access application to upload a request - post a request on app
Iteration: 1, created: February 20 by A. Mulla.
Primary actor: User (Tenant, Landlord, Student, Residence Hall)
Goal in context: To allow user to request a task to be completed
Preconditions: Request fields must be entered: Date, Time, Duration, Task
Trigger: The user requires a task to be completed

Scenario:

1. The user logs on the Dorm.ly app
2. The user enters his or her user ID and password. If "Auto Login" accepted, the system will bypass login page
3. The system will load in the home screen when credentials are validated
4. The user will select the "Create Task" button from the home screen or the options menu bar.
5. The system will provide form fields to be filled in.
6. The user will fill out the "Date" field if the user is requesting for the same day or future date.
7. The user will fill out the "Time" field if the user is requesting for a specific time of day.
8. The user will fill out the "Duration" field if the user is requesting for services that require to be completed immediately or for a length of time.
9. The user will fill out the "Task" field to include the required service(s) or include additional details on task(s) to be performed and/or payment methods.
10. The system will prompt the user "Please Complete" if the user does not include any text in the "Task" field.
11. The user will select the post button.
12. The system will post the task and prompt an alert that the user that a pending post has been created.
13. The system will include a "Pencil" and "X" button on the top right of the prompted alert for the user to edit or cancel the created request.

Exceptions:

1. ID or passwords are not validated - see use case **Login Validation**
2. Date, Time, Duration fields are not filled in - system automatically replaces fields with "TBD"; see use case **Task Availability**.
3. User fails to include task - system displays appropriate error message - see use case **Task Description**.
4. User wants to remove or edit the post - see use case **Task Completion**.
5. Task is overdue - see use case **Task Completion**.

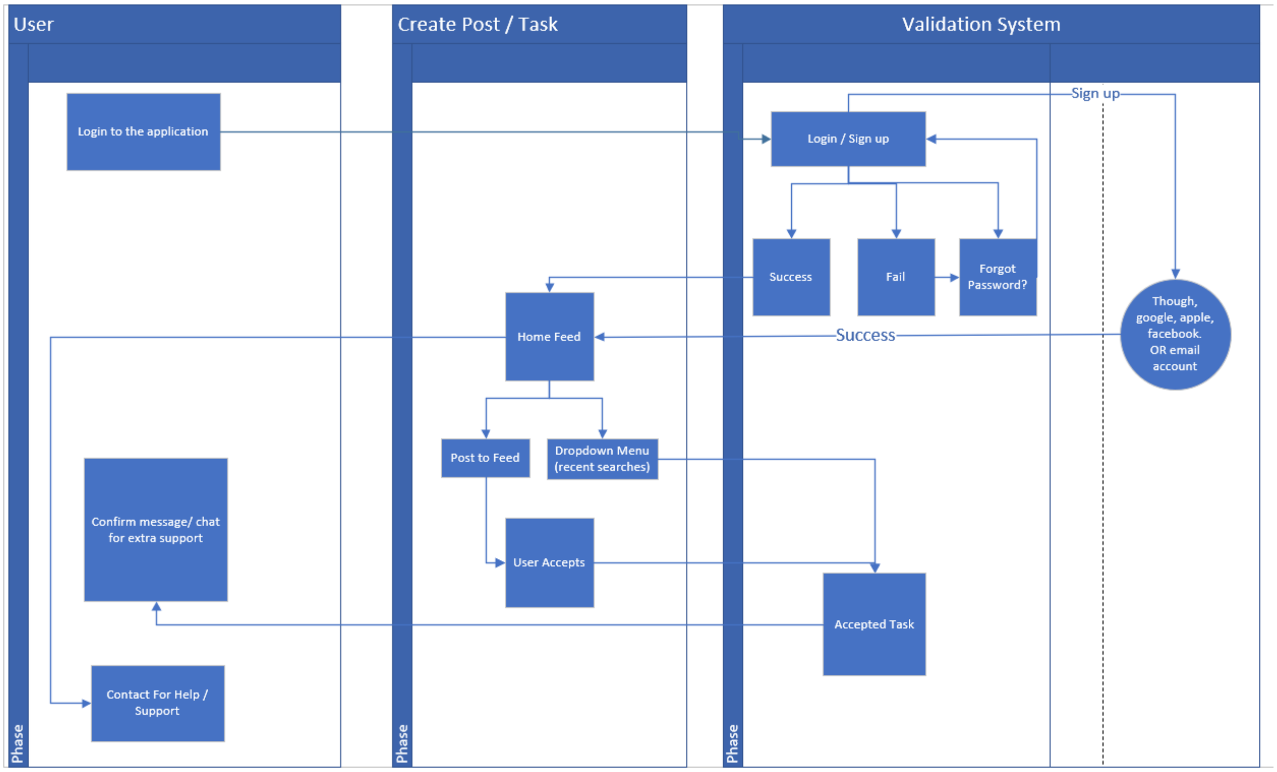
Priority: High priority, to be implemented with basic functions.
When available: First increment.
Frequency of use: Frequent
Channel to actor: Via Mobile or PC device and Internet connection
Secondary actors: Users

Channels to secondary actors:

1. Users: Mobile or PC device

Open issues:

1. Will the system appropriately filter age restricted or unsafe task(s)?
2. What liability risks could be present and how will it be mitigated?
3. Will there be a purchase and seller protection for tasks that involve funds transactions?
4. Is the system able to confirm if the task is completed to the requested user's satisfaction? What standards are required to complete a task.



5. Other Nonfunctional Requirements

5.1 Performance Requirement

Nonfunctional Requirements list				
Requirement ID	Requirement title	Short Description	Priority	Requester
NFR01	Mobile compatibility	The app needs to be of running on devices such as iOS or Android.	High	IT Administrator (Shana Duke)
NFR02	Video calling	The system should have a video chatting method for users that want to talk to each other face-to-face.	Normal	Programming Lead (Ameen Mulla)
NFR03	Cybersecurity	The system should be able to prevent against malicious hackers or DDoS attacks.	High	Investor (Zayden Donald)
NFR04	Web browser compatibility	Users that want to access the app on their PC or Mac should be able to do so.	Normal	IT Administrator (Shana Duke)
NFR05	Voice-command	Users should be able to use Alexa/Siri/Google to navigate the application.	Normal	IT Administrator (Shana Duke)
NFR06	Multilingual	The app needs to be translated into languages other than English.	High	IT Administrator (Shana Duke)
NFR07	AdSense	The app will contain ads for the free version. Purchase of the app will remove ads.	Normal	Investor (Zayden Donald)
NFR08	Background App Refresh	The app will be allowed to refresh data by the device.	Normal	IT Administrator (Shana Duke)
NFR09	Language Filter	The app will be monitoring (continuous update & learning) to allow for filtering of profanity.	Low	IT Administrator (Shana Duke)
NFR10	Content Filter	The app will monitor and filter inappropriate content posted on	High	IT Administrator (Shana Duke)

		chat message or anywhere else on the map.		
NFR11	Request Filter	The app will monitor and track illegal or inappropriate requests by all levels of users to prevent illegal/unwanted activity.	High	IT Manager (Derek Regan)

Use Cases			
Use Case Name	List of Related Requirements ID	Actor (s)	Brief Description
Open application	FR 05	Users	The actor will click on the application. The system will respond by opening the application to the home page.
Translate Language/Change Language	FR 10	Users, Customer Support Agent	The actor will be able to see in the options setting menu a list of languages to select from. The actor will press the selected language on their screen. The selection will translate text throughout the entire application. The setting will save the preference until it is changed again by the actor.
Add Member	FR 03	Users	The actor can add new members to the circle by pressing the +Add button. The actor can view their profile including history and rating. The actor can manage members within the circle.

Post a Task/Request	FR 08	Users	The actor will create a task or request and post in the chatroom or status update. In response the system will notify or alert other users to view the request. Other actors will view the task and have the option to complete the task or not.
Search Button	FR 11	Users	The actor will be able to press this button, in response the system will show a drop-down function to related or previous searches. Also giving the option to further search people who are available to complete the task or post a new task.
Message User Button	FR 04	Users	The Message button will open a chat box or give a video call option that will then allow the actors to get in contact with each other. The system will alert the other actors if they miss a call or receive a message.
Pop-up Notifications/Alerts	FR 02	Users	The actor can enable and disable push notification or alerts by pressing or sliding the button from within the app or from their mobile device settings. Functionality will be mirrored in both locations. The actor has the option of setting a

			timer to temporarily mute notifications.
Payments and Transaction of Funds	FR 09	Users, Payment API	The actor will see the option of sending funds by pressing the Pay Now button. The button will be available if there is an outstanding balance. The application will connect to a Third-Party solution to verify credentials. The recipient will be notified of the transaction.
Rate User	FR 01	Users	The actor will rate other actors based on their performance or quality after performing a task. In response the system will update the rated actor's total 'score' out of 5.
Call Customer Support Button	FR 06	Users, Customer Support Agent	The actor will push a button on the application. In response the system will automatically connect the actor with a customer support agent where they can message in chat or talk over the phone.
Report User Button	FR 06	Users	The actors can report users violating rules to their respective administrator after pressing a button. They will be able to type a response on why they are reporting the user. This button will appear

			as an option after rating the user.
Exit Application	FR 05	Users	The actors can exit the application, the system will respond by asking to run in the background or close the application completely.

6. Other Requirements

Appendix A: Glossary

SRS	Software Requirements Specification
DM	Direct Message
DDoS	Distributed Denial of Service
OS	Operating System
APP	Application
IT	Information Technology
FR	Functional Requirement
NFR	Non-Functional Requirement
FAQ	Frequently asked questions
NFC	Near field communication
C#	High level programming language
RAM	Random access memory
Wi-fi	Wireless fidelity
USB	Universal Serial Bus
HTTP	Hypertext transfer protocol
FTP	File transfer protocol
VPN	Virtual private network
IP	Internet protocol
API	Application programming interface
PHP	Personal home page
ToS	Terms of service
HTML5	Hypertext markup language 5
TBD	To Be Discussed

Appendix B: Analysis Models

TBD: Part B

Appendix C: Stake Holders

- Section 6
- Appendix B

Richard McCreggor	Customer	External	rickylandlord@gmail.com	Operational	High
Janice Ryerson	Customer	External	janiceresidirector@ymail.com	Operational	Medium
Mathew Nava	Marketing officer	Internal	mathew.nava@live.ca	Operational	Medium
Ellie Coffey	Marketing manager	Internal	ellie.coffey@yahoo.ca	Executive	High
Shanna Duke	IT administrator	Internal	shanna.duke@hotmailcom	Operational	Low
Derek Regan	IT manager	Internal	derek.regan@tech.ca	Executive	High
Zayden Donald	Investor / Venture Capital	External	zayden.donald@gmail.com	Executive	High
Elyse Millar	Customer Support	Internal	elyse.millar@talktous.com	Operational	Low
John Tran	Junior Programmer	Internal	john@my.centennialcollege.ca	Operational	Low

Joshua Soriano	Senior Programmer	Internal	josh@my.centennialcollege.ca	Executive	High
Ameen Mulla	Programmer Lead	Internal	ameen@my.centennialcollege.ca	Executive	High
Lester Dan Goze	Junior Programmer	Internal	lester@my.centennialcollege.ca	Executive	Medium

Appendix D:

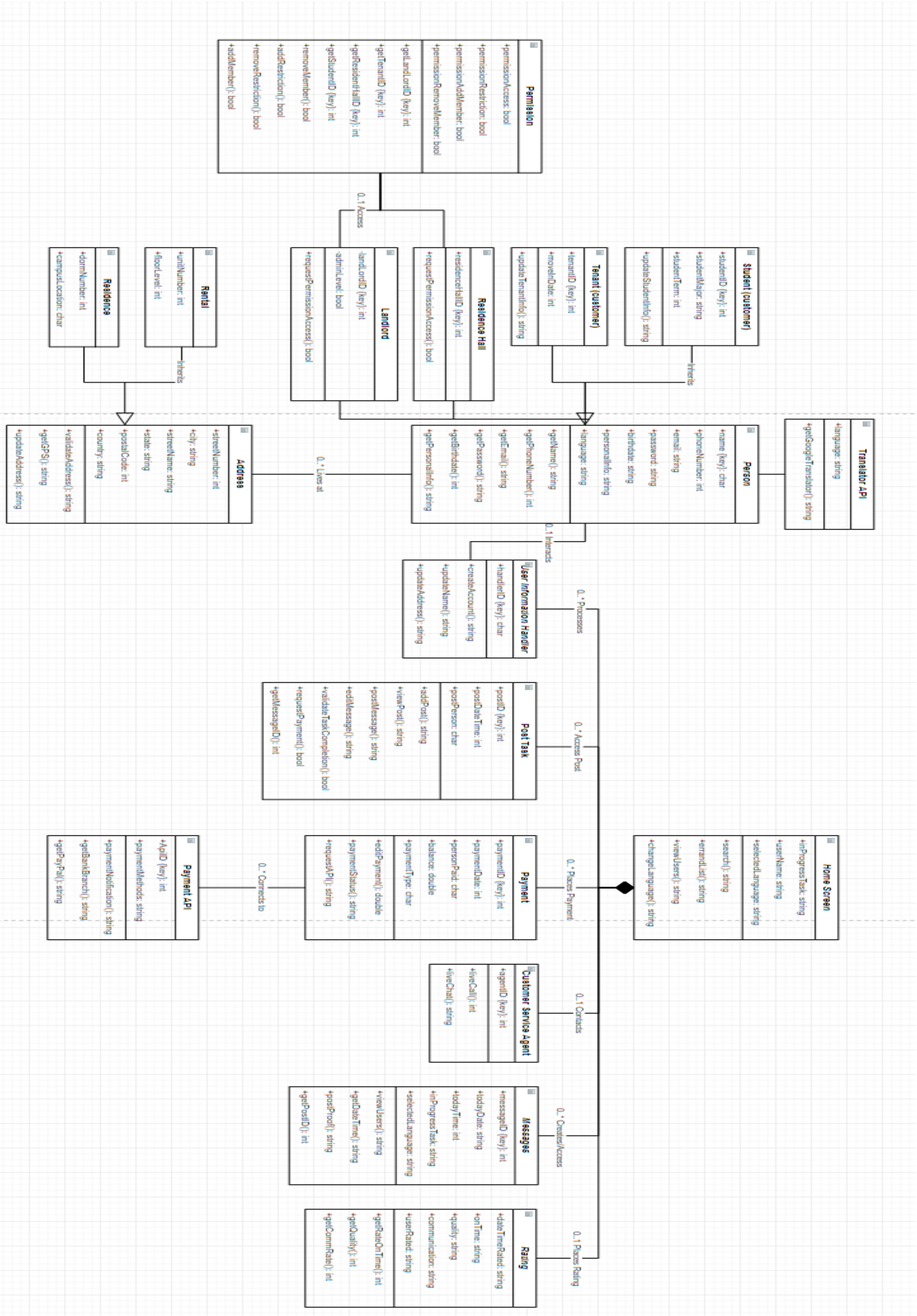
Interview Questions		
Question	**Stakeholder position**	**Answer**
1) In the first iteration, who will be the targeted audience?	Marketing Manager	To students: School dormitories and student housing. Landlords and small properties (1-3 units max).
2) How often will the application be used?	Customer	Landlord: Weekly basis for minor maintenance Residence Hall: Daily to ensure garbage bins are emptied and replaced with new bag.
3) Are there any restrictions on errands which would not be allowed?	Investor	Request to purchase age restricted items such as alcohol, cigarettes, and cannabis. To reduce risk of liability.
4) Are there any privacy issues to look for with the service?	IT Manager	Agreement and verification of identity.
5) How do you plan on scaling the project?	Marketing Officer	Advertisements through school (college & university residence/dorms/housing). Supporting multiple languages across the world. Extending support to small businesses to handle certain maintenance.
6) Who are your primary competitors? What are you doing differently?	Marketing Officer	A few grocery apps come to mind: • Postmate • Uber Eats • Facebook

7) Is there a subscription fee or additional cost to using the app?	Investor	There is no subscription fee, but there will be a premium option where no ads are displayed. Services are not outsourced to unknown users. Sponsorships from schools or businesses to keep app operational. As the app evolves and more features are implemented then offering subscription fees may become optional.
8) Will there be a refund system if the end user is not satisfied and how does it work?	Marketing Officer	There's no refund, but a trial version is offered in the beginning where a customer has 30 days to explore the app prior to committing on the purchase. The app is free although there will be advertisements.
9) How will these apps be integrated and useful for people who are busy	IT Manager	We will be making sure to keep the app - on the go friendly by implementing it on newest OS features like (Apple Watches).
10) How will customer service operate, 24/7?	Customer Support	We will have a live chat feature and have an email for questions.
11) How will the app be developed? What software languages are you using to code the application?	Programmer	High level languages such as JavaScript & C#. React for UI integration on the mobile app. HTML5, JavaScript, PHP for implementing a desktop version in the future.
12) What APIs will the application be using?	Programmer	Text-to-speech, weather, Google maps, web search.

13) What are the options users have to post an ad	Marketing Manager	Users can post an ad, or they can search for their ad and see members who are available at the moment to do them.
14) Who is designing the application interface?	Programmers	A part of the programming team will design the UI.
15) How many tasks can someone accomplish in a day? Is there a limit?	IT Manager	There won't be a limit, as long as each task is accomplished and certified, another one can be accepted. As long as it meets the time requirements.
16) How much space will the application take? Example on handheld devices to optimize memory	Programmer	We will target for around 50MB, similar to other social media applications like Facebook.
17) What OS will the app be developed for?	Programmer	The application will be developed for Android Nougat (v7.0) and higher, and iOS 8.0 and higher for Apple devices.
18) How will you plan your development cycles? Will you implement an AGILE methodology or SCRUM meetings?	Programmer	AGILE methods will be implemented as it will provide a direction in which the team can work on. We can hold SCRUM meetings on specific days to compile each programmer's work.
19) Is there a terms and conditions/privacy policy in effect?	Marketing Manager	Users of the application must agree to a ToS to use the application. This is added protection for users and the development team.
20) Will errands be accepted after work hours? What is the schedule for the app	IT Manager	Yes, errands can be accepted but have more strict rules as limitations for customer safety and security as it will be in the night time.

Appendix E:

6.1.1 CRC Diagram



Refer to class_responsibilites_collaboration_1.pdf file

6.1.2 CRC Cards

<table>
<tr><td colspan="2">Permission</td></tr>
<tr><td colspan="2">Description: Provides administrative access</td></tr>
<tr><td>Responsibilities:
• Validate members in local hub
• Get restrictions in local hub</td><td>Collaborators:
• None</td></tr>
</table>

<table>
<tr><td colspan="2">Residence Hall</td></tr>
<tr><td colspan="2">Description: Sets user permissions Residence Hall</td></tr>
<tr><td>Responsibilities:
• Can make requests
• Create permissions in local hub</td><td>Collaborators:
• Students (customer)
• Permission</td></tr>
</table>

<table>
<tr><td colspan="2">Student (customer)</td></tr>
<tr><td colspan="2">Description: Sets user permissions to student (customer)</td></tr>
<tr><td>Responsibilities:
• Update student information</td><td>Collaborators:
• None</td></tr>
</table>

Landlords	
Description: Sets user permissions to Landlord	
Responsibilities: • Create permissions in local hub • Add/remove restrictions for tenants	Collaborators: • Permissions • Tenant (customer)

Tenants (customer)	
Description: Sets user permissions to student (tenant)	
Responsibilities: • Update tenant information	Collaborators: • None

Rental	
Description: Defines living arrangement	
Responsibilities: • Collects building information	Collaborators: • Landlord • Tenant (customer)

<table>
<tr><td colspan="2">Residence</td></tr>
<tr><td colspan="2">Description: Defines living arrangement</td></tr>
<tr><td>Responsibilities:
• Collects building information</td><td>Collaborators:
• Resident Hall
• Student (customer)</td></tr>
</table>

<table>
<tr><td colspan="2">Translator API</td></tr>
<tr><td colspan="2">Description: Changes to the user’s preferred language.</td></tr>
<tr><td>Responsibilities:
• Get selected language</td><td>Collaborators:
• User Information Handler</td></tr>
</table>

<table>
<tr><td colspan="2">Person</td></tr>
<tr><td colspan="2">Description: Defines basic user information</td></tr>
<tr><td>Responsibilities:
• Get/Update Name
• Get/Update Phone number
• Get/Update Email
• Get/Update password
• Get birthdate
• Get/Update personalized info (interests, hobbies)
• Request selected language</td><td>Collaborators:
• User Information Handler
• Translator API</td></tr>
</table>

<table>
<tr><td colspan="2">Address</td></tr>
<tr><td colspan="2">Description: Defines user address</td></tr>
<tr><td>Responsibilities:
• Get/Update Address
• Validate user address</td><td>Collaborators:
• Person</td></tr>
</table>

<table>
<tr><td colspan="2">User information handler</td></tr>
<tr><td colspan="2">Description: Defines user address and living arrangement</td></tr>
<tr><td>Responsibilities:
• Creates user account
• Create living arrangement details</td><td>Collaborators:
• Rental
• Residence
• Person
• Address</td></tr>
</table>

<table>
<tr><td colspan="2">Post Task</td></tr>
<tr><td colspan="2">Description: Allows users to request or accept tasks.</td></tr>
<tr><td>Responsibilities:
• Get/update errands or tasks
• Validate completion of tasks
• Request payment transactions</td><td>Collaborators:
• Messages
• Payment</td></tr>
</table>

Payment API	
Description: Fulfills payment transactions for tasks	
Responsibilities: • Provides options for payment methods such as visa, credit card, bank or PayPal • Process transaction	Collaborators: • None

Payment	
Description: User can create funds transactions after discussing with other users, and also access bank information	
Responsibilities: • Request payment API access • View payment status	Collaborators: • Messages • Payment API

Customer Service Agent	
Description: Connects users to calls and chats with customer service agents	
Responsibilities: • Create a live call • -Create a live chat	Collaborators: • User Information Handler • Post

Messages	
Description: Allows users to communicate with other users (e.g., require additional information for current task/ payment details).	
Responsibilities: • Create messages to send to other users • Update messages • View messages	Collaborators: • Payment • Post

Rating	
Description: Gives users the option to rate one another after a completed task.	
Responsibilities: • Updates rating information	Collaborators: • Post • Messages

Home Screen	
Description: Displays the dorm.ly main menu screen, where users can then decide to Post or Accept available tasks.	
Responsibilities: • Collect user information • Create tasks and errands • Message other users • Connect with customer support • Create payment transactions	Collaborators: • User information handler • Post • Payment • Customer Service Agent • Rating

6.2 State Transition Diagram

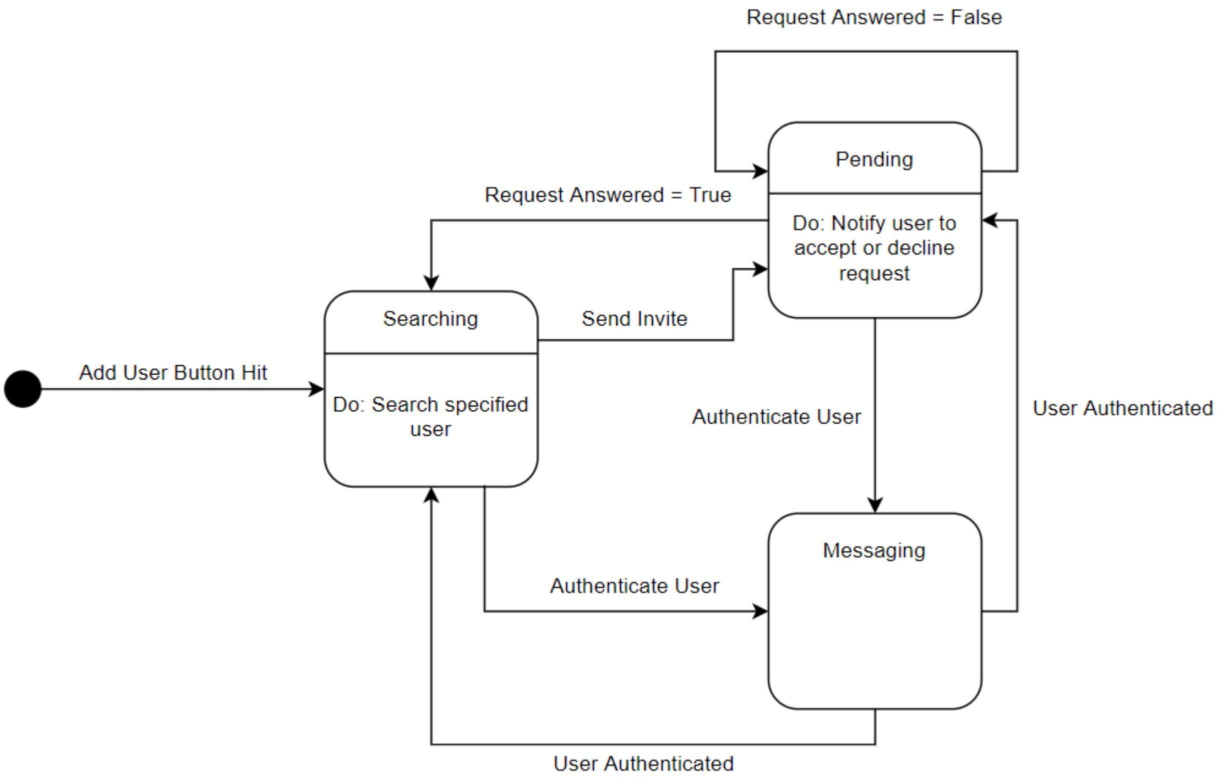
6.2.1 Add User State Diagram

Triggering Event: User presses the “Add User” Button
States:

- Searching
- Pending
- Messaging

Transitions:

- Send invite
- Request Answered = False
- Request Answered = True
- Authenticate User
- User Authenticated



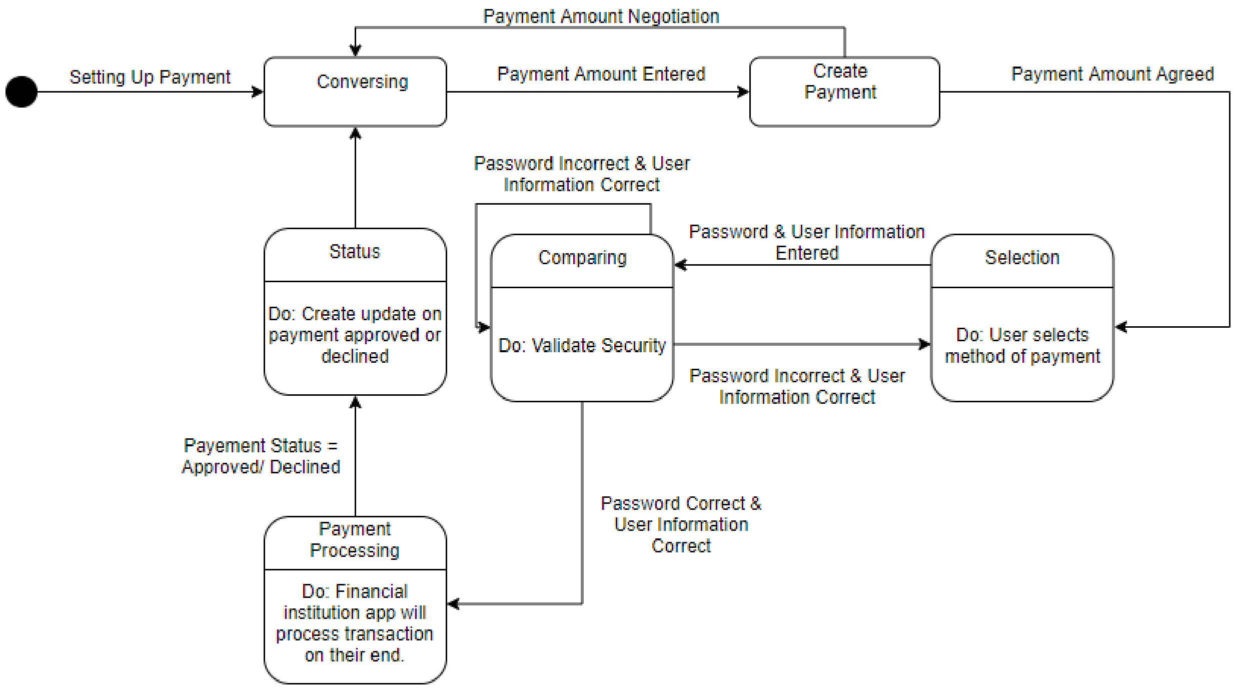
6.2.2 Payment State Diagram

Triggering Event: User presses the "Set Up Payment" Button
States:

- Conversing
- Create Payment
- Selection
- Comparing
- Payment Processing
- Status

Transitions:

- Payment Amount
- Password & User Information Correct
- Password & User Information Incorrect
- Payment Status


Payment Amount Negotiation
Setting Up Payment
Conversing
Payment Amount Entered
Create Payment
Payment Amount Agreed
Password Incorrect & User Information Correct
Status
Do: Create update on payment approved or declined
Comparing
Password & User Information Entered
Selection
Do: Validate Security
Do: User selects method of payment
Password Incorrect & User Information Correct
Payement Status = Approved/ Declined
Password Correct & User Information Correct
Payment Processing
Do: Financial institution app will process transaction on their end.