Requirement Elicitation

InboundRX Capstone Team

Brett Chafin, Jason Custodio, Jason Brophy, Paul Huynh, Luke Kwak, Cher Moua, Thaddeus Sundin

1. Introduction
    a. Purpose of the system:
    Paulsen's iOS app serves to connect users to Paulsen's Pharmacy and give them access to exclusive deals and services. The app will communicate with Paulsen's web server, along with several Estimote beacons placed within the pharmacy, which will allow users who have signed in to their Paulsen's account to collect points just by visiting. The app will allow users to redeem these points for exclusive vouchers, which can be activated and used within the pharmacy. The app contains information about deals and sales within the pharmacy, and additionally serves as a way for users to securely request prescription refills.

    b. Scope of the system:
    The app will be able to securely connect to Paulsen's web server any time the user has the app open on their device and said device has a data connection. The app and web server will exchange user information, as well as information about deals within the store. The app will connect via Bluetooth to beacons that are placed and activated in Paulsen's Pharmacy. Connections with beacons can occur whether the user has the app open or not.

    c. Objectives and success criteria of the project:
    For this project to be successful, it should meet the following criteria:
        1. The application is fully tested and deliverable by the second week in March
        2. The application has the following features:
            a. A user registration and login system that allows persistence of user information.
            b. Full beacon integration, which acts as the primary driver for the points system, and allows pushing notifications to users phones operating the application.
            c. A persistent points system which allows for collection and redemption of points for store items.
            d. A page to view the current deals at the store
            e. The ability to email the pharmacy and order a refill of a prescription for either pickup or delivery. This must be secured for HIPAA compliance.
        3. The user interface is easy to read, and easy to navigate, specifically for the elderly.
        4. The application interfaces with the web server to store and update persistent information:, registration, login info, point totals, current deals, and beacon information.
    d. Definitions and acronyms:
        i. Account: Each user may have an account with Paulsen's. This account will be tied to that user's email address, and will contain information such

as their accumulated points, their address and their prescription information. Account information is stored within Paulsen's web server.

ii. App/Application: The iOS application to be created as per this document that a user will download from the Apple App Store and have installed on their phone.

iii. Beacon: An Estimote beacon which will be in fixed locations within the pharmacy. Each beacon sends out Bluetooth frequencies that can be received by user's mobile devices and which will trigger specific events within the application.

iv. HIPPA: In the United States HIPAA,Health Insurance Portability and Accountability Act, is an act that gives protection to each person medical information.

v. iOS: The operating system Apple mobile devices and tablets run on.'

vi. InboundRx: The marketing company that commissioned the development of the application.

vii. Pharmacy: Paulsen's Pharmacy; The physical store the application is to be used to promote.

viii. Point: A digital currency that is awarded to the user when they enter the pharmacy. Points are tied to each user's account, and they can be redeemed for vouchers within the app.

ix. User: A customer of Paulsen's who has or will download the app onto their mobile device. Each user may also have an account with Paulsen's which contains their information.

x. Voucher: A digital ticket that can be purchased within the app using points. Vouchers can then be "activated", at which point they can be used within the pharmacy and they will expire after 5 minutes.

xi. Web Server: Paulsen's web server which consists of a database containing user account information, current deals and promotions and beacon information. The app will communicate with the web server to authenticate users and pull and store their information, and to retrieve information about deals and promotions.

e. References:

i. Estimote Beacons, http://www.estimote.com/

ii. HIPAA, http://www.hhs.gov/hipaa/

2. Proposed System

a. Overview

This section is intended to provide details about Functional Requirements and Nonfunctional requirements of the proposed system, and provide System models created during the process. This will cover how the user interacts with the Application, how the Application interacts with the Web Server/Beacons, and vice versa

b. Functional requirements

Use Cases

Name: EnterBeaconRange
Initiating Actor: User
Entry Condition:

- The user has the application installed on the phone, and has already registered, and has the application running in the background, while logged in. The beacon is activated with information transmitting.

Exit condition:

- The user terminates the notification.

Event Flow:

Actor Steps	**System Steps**
1.The user enters outer range of a specific beacon	
	2.The application registers the beacon signal
	3.The application deciphers the beacon information
	4.<includes> PushUserNotification

Name: CreateAnAccount
Initiating Actor: User
Entry Condition:

- The user has the application installed on the phone
- The user does not have an account

Exit condition:

- User's account in created and home screen is displayed

Event Flow:

Actor Steps	System Steps
1.User opens app.	
	2.The app welcomes the user, displays the sign in/create an account page, and awaits response from user.
3.User selects to create an account	
	4.The app displays the account creation form for the user to fill out.
5.User fill out account creation form and hit 'Submit'.	
	6.The app takes this information and sends it to the Web Server for processing/storing. <extends> NoWebServerConnection
7.Web Server sends information to app.	
	8.App goes to the homescreen and welcomes the user to the app. <extends> NoWebServerConnection. <extends> AccountAlreadyExists

Name: UpdateBeacon
Initiating actors: Web-server
Entry condition:

- There is a User with the app

Exit Condition:

- The App receive the beacon info.

Event Flow:

Actor Steps	System Steps
	1.The App is running
2.Web Server notifies App that it needs to update beacon info.	
	3.App pulls beacon info from the Web-Server

Name: OpenHomeMenu
Initiating Actor: User
Entry Condition:

- The user has the application downloaded, and is on the home screen

Event Flow:

Actor Steps	System Steps
1.The user clicks the Menu button on the home screen	
	2.The Application pulls up the menu

Name: ViewDeals
Initiating Actor: User
Entry Condition:

- The User is on the Menu from the Home Screen

Exit Condition:

- The User closes the app
- The User leaves the Daily Deal page

Actor Steps	System Steps
1.User clicks on daily deals	
	2.The app will fetch the daily deals from a the web server. <extends> NoWebServerConnection
3.The web server sends the information about the deals	
	4.The app stores the deals <extends> NoCurrentDeals
	5.The app opens the daily deal page with the first deal displayed
6.The user presses the left or right arrow to view the previous deal or next deal.	
	7.The application stays on the same page but displays the correct

Name: ViewHistory
Initiating Actor: User
Entry Conditions:

- The user has opened the Menu in the application (Case: OpenMenu)

Exit Condition:

- The user clicks the back button to go back to the home screen.
- The user exits the application.

Event Flow:

Actor Steps	System Steps
1.The User clicks the history button on the menu	
	2.The Application opens the History page
	3.The Application displays the stored history timeline on the screen.
4.The User clicks on one of the events on the timeline	
	5.The Application displays a popup of that timeline events information
6.The user clicks outside the popup	
	7.The popup closes

Name: ViewPresciptionPasswordValidation
Initiating Actor: User
Entry Condition:

- The User is signed into the application
- The User tries to view their prescriptions

Exit condition:

- The User is allowed access to the ViewPresciption page

Event Flow:

Actor Steps	**System Steps**
1.The User attempts to navigate to the ViewPrescription page within the App.	
	2.Before allowing access, the App displays a password field and asks the user to input their password.
3.The User enters their password into the field and hits the 'submit' button. <extends> IncorrectPassword	
	4. App displays ViewPrescrition page

Name: IncorrectPassword
Initiating Actor: Web Server
Entry Condition:

- The App has sent the User's password input to be validated by the Web Server

Exit condition:

- The App displays 'incorrect password' notification to the User and re-prompts them for input.

Event Flow:

Actor Steps	System Steps
1.The Web Server determines that the User password input is incorrect.	
2.The Web Server sends an error notification to the App, saying that the password was incorrect.	
	3.The application displays an 'incorrect password' notification to the user and re-prompts the User for input. <extends> ForgotPassword

Name: ForgotPassword
Initiating Actor: User
Entry Condition:

- User clicks on "Forgot Password" button on sign in page

Exit condition:

- App emails password information

Event Flow:

Actor Steps	System Steps
1.User inputs email	
	2.App checks for user info on Web Server. <extends> NoWebServerConnection
3.Web Server validates and sends password information to App.	
	4.App sends email to User about account password

Name: ReorderPrescription
Initiating Actor: User
Entry Condition:

- The User is signed into the app

Exit condition:

- The App has emailed the pharmacist and notified the User.

Event Flow:

Actor Steps	**System Steps**
1.The User navigates to the ViewPresciption page.	
	2.<includes> ViewPresciptionPasswordValidation The App requests the User's prescription data from the Web Server. <extends> NoWebServerConnection
3.The Web Server sends the App the requested prescription information.	
	4.The App displayed the User's current prescriptions.
5.The User selects what prescription they want to reorder, and hits the reorder button.	
	6.The App emails the pharmacy with the necessary reorder information.
	7.The App displays a notification informing the User the Email has been sent, and that they will be Emailed when their prescription is ready to be picked up. <include> PushUserNotification <extends> DeliverPrescriptions

Name: UserRevisitsStore
Initiating Actor: User
Entry Condition:

- User has app in background
- User already has a registered account
- User walks into the store

Exit Condition:

- User close store notification on his phone

Event flow:

Actor Steps	System Steps
1.User walks into the store	
	2.App checks with Web Server for visit count. <extends> NoWebServerConnection
3.Web server responds with info.	
	4.If count > 1, App push a welcome back notification and reward points. <includes> PushUserNotification
	5.App then stores user's earned reward points into the web server.

Name: UserViewsAccountInformation
Initiating Actor: User
Entry Condition:
- User already has a preexisting account
- User has the app on their phone

Exit Condition:
- Users' points are displayed on the home screen of the app

Event flow:

Actor Steps	System Steps
1.User opens app.	
	2.App requests user's points and daily deals from web server. <extends> NoWebServerConnection
3.Web server responds with info.	
	4.The app displays this info on the home screen.

Name: DeliverPrescriptions
Initiating Actor: User
Entry Condition:
- User sends reorder prescription information to App

Exit Condition:
- User confirms order

Event Flow:

Actor Steps	System Steps
1.User chooses prescriptions to be delivered instead of pick up	
	2.App checks Web Server for saved Address. <extends> NoWebServerConnection
3.Web Server sends Address to App <extend> AddressNotSaved	
	4.App sends success notification <include> PushUserNotification

Name: AddressNotSaved
Initiating Actor: Web Server
Entry Condition:

- User has not saved their address in the app.
- User is in the prescription delivery window of the application

Exit Condition:

- User enters in the address when prompted, or cancels out.

Event Flow:

Actor Steps	System Steps
1.Web Server tells app that address is not saved	
	2.App prompts user to enter in an address or cancel out of the screen.
3.User can save address	
	4.App sends address to Web server.
5.Web Server saves address	

Name: RedeemVoucher
Initiating Actor: User
Entry Condition:

- User must have enough points to redeem.
- User already has an account

Exit Condition:

- User shows cashier the coupon and the cashier manually deducts the cost.

Event Flow:

Actor Steps	System Steps
1.User goes to the redeem points page or presses the points number on the front page. <extends> NoWebServerConnection	
2.Web server looks at the user's point value and communicates with the app on what is available to be redeemed.	
	3.App presents redeem page. Unredeemable Vouchers are greyed out.
4.User presses the reward that they want to redeem.	
5.Web server deducts the points from the user's rewards points value	
6.User redeems a voucher <extends> ActivateVoucher	

Name: ActivateVoucher
Initiating Actor: User
Entry Condition:
- User must have redeemed a voucher

Exit Condition:
- Voucher timer runs outs
- Voucher then removed

Event Flow:

Actor Steps	System Steps
1.User clicks “activate” button on Voucher	
	2.App sends pop up to “Accept” or “Decline” activation
3.User clicks Accept	
	4.App starts countdown timer of 5:00 min

Name: PushUserNotification
Initiating Actor: User
Entry Condition:
- App is running in background
- App is pinged to notify user

Exit Condition:
- User closes notification

Event Flow:

Actor Steps	System Steps
	1.App sends notification to User
2.User can close or ignore notification	

Name: NoWebServerConnection
Initiating Actor: Web Server
Entry Condition:
- User uses app feature that needs to access web server

Exit Condition:
- Send notification for failed attempt

Event Flow:

Actor Steps	System Steps
	1. App tries to connect to Web Server
2.Web Server does not respond	
	3.App sends error notification <include> PushUserNotification

Name: AccountAlreadyExists
Initiating Actor: Web Server
Entry Condition:
- User tries to create an account

Exit Condition:
- Send notification to use different account id

Event Flow:

Actor Steps	System Steps
	1. App checks with Web Server if account details are already in database <extends> NoWebServerConnection
2.Web Server tells app that account exists	
	3.App sends use different id notification <include> PushUserNotification

Name: UserSigningInTheApp
Initiating Actor: User
Entry Condition:
- User opens app

Exit Condition:
- App displays home page

Event Flow:

Actor Steps	System Steps
1.User selects "Sign In" button	
	2.App displays sign in page
3.User puts in user details	
	3.App shows home page <extends> IncorrectPassword

Name: NoCurrentDeals
Initiating Actor: Web Server
Entry Condition:
- Web Server returns no deals

Exit Condition:
- The user exits the daily deals page

Event Flow:

Actor Steps	System Steps
	1.App goes to the daily deals page and displays a message stating there are currently no deals

c. Nonfunctional requirements
   i. Performance
      1. Communicating with the web server should not take a long time
      2. Properly pushes the right notification when user is in beacon's range
      3. Properly load users information every time they get on the app
   ii. Security
      1. Having a secure way for the user to refill their prescription on the app without violating HIPAA

2. Keeping the user's information private as possible

iii. Design Constraints
1. App has to communicate with the web server
2. App has to be simple and easy for the elderly to use
3. App gives user a way to refill their prescription while still protecting their information

iv. Usability
1. Easy for all users to learn how to use
2. The appearance of the objects on the app should be bigger for users to spot

v. Error Handling
1. App should prompt user to retype their input until correct

vi. Reliability
1. App should be up 99.99% of the time
2. Properly informs pharmacy when the user's want their prescription refill, for pick-up or delivery.
3. User's can't lose their redeemed vouchers if they have not activated them.

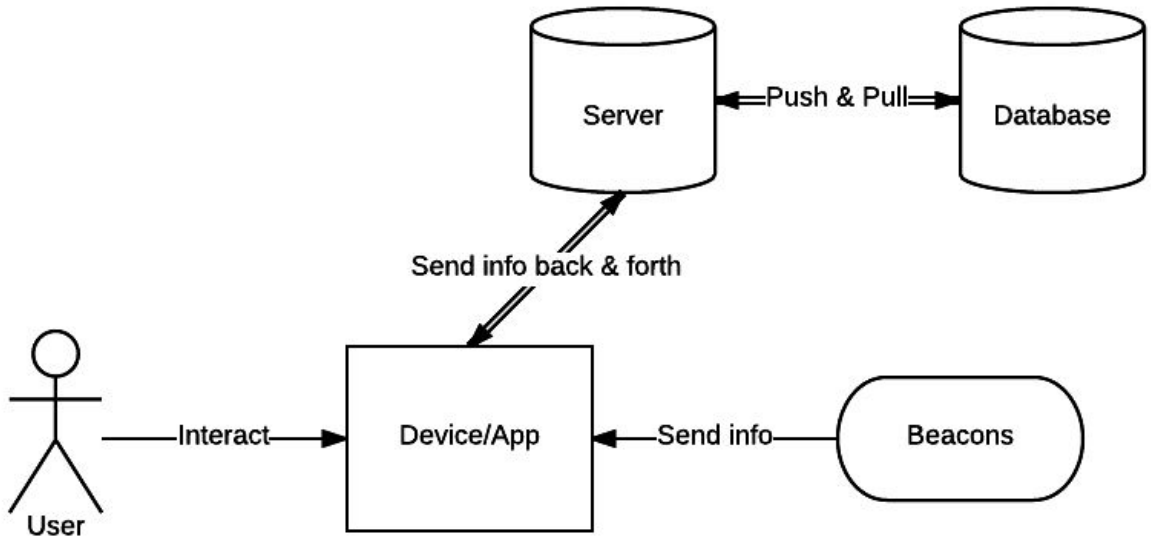
vii. Operation
1. InboundRx manages the data of the app through the web server

viii. Interface
1. App will be an iOS application
2. Uses SDK from Estimote to help communicate with beacons
3. Pulls and push users data to web server

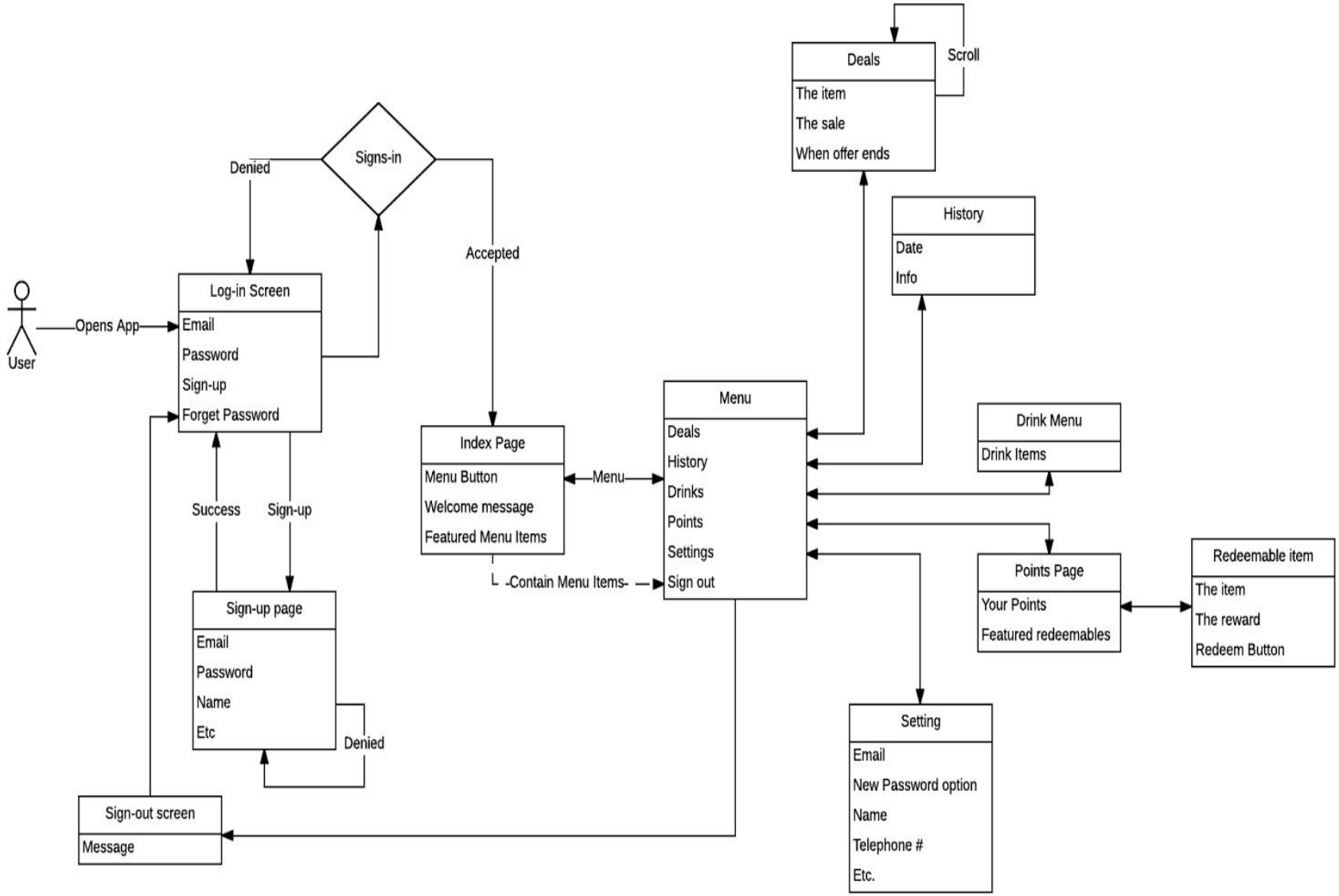
ix. Legal
1. Lawsuit may occur if violating HIPAA

d. System models

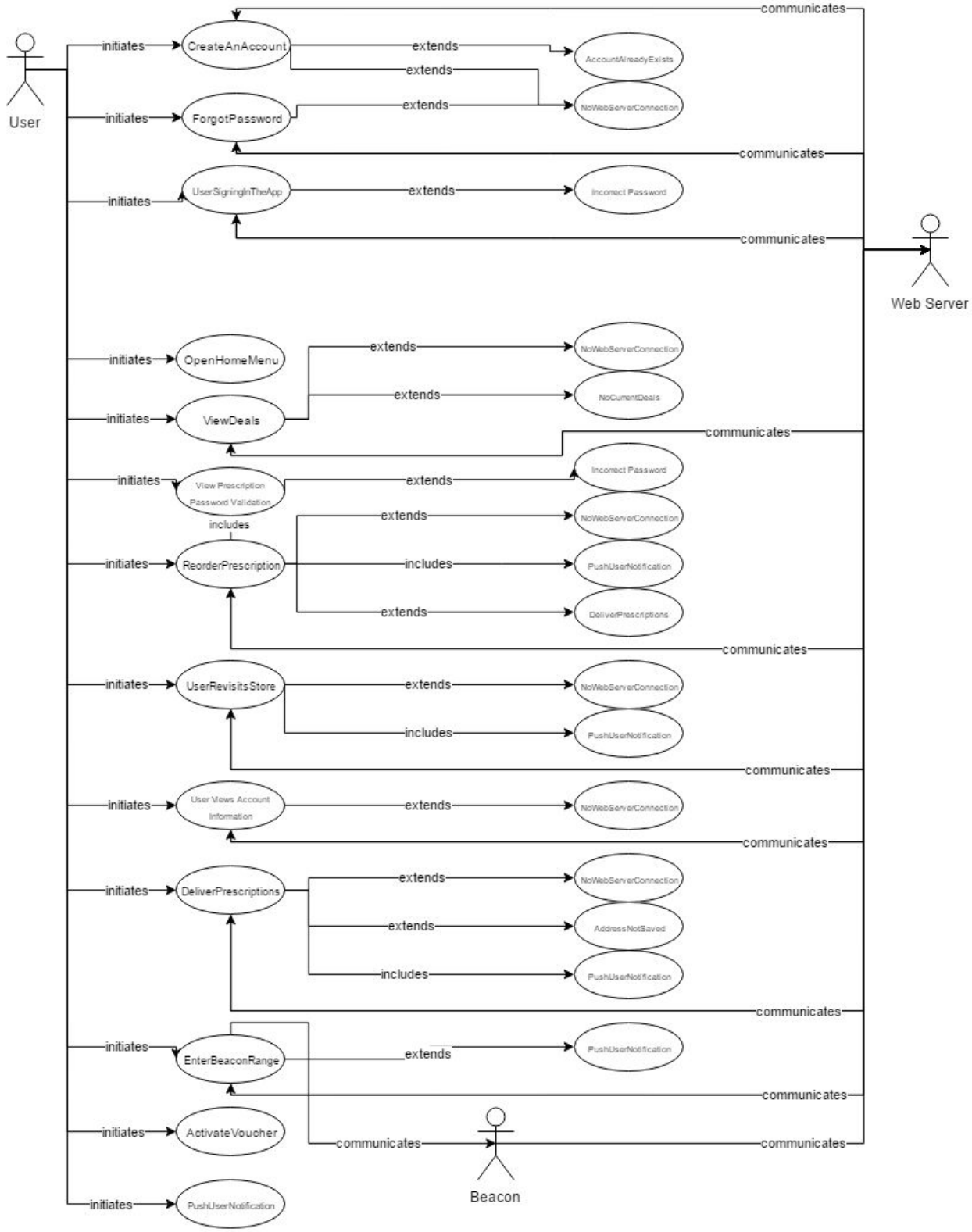
i. Overall System

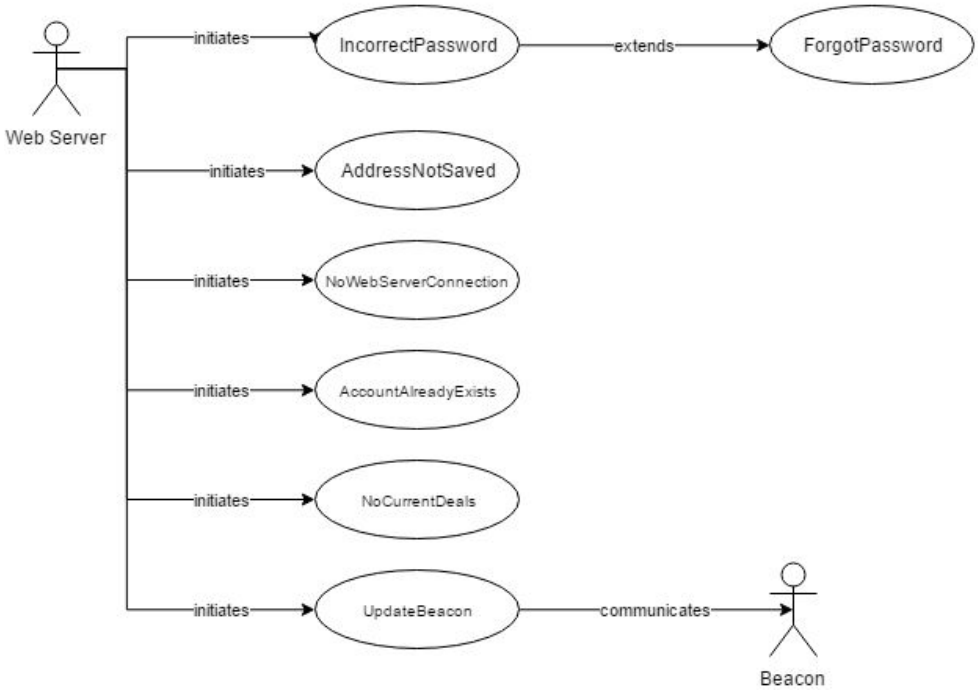


ii. Rough ER Flow graph of App

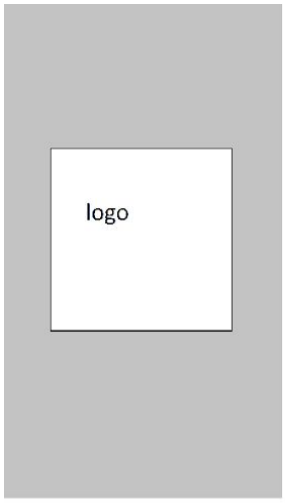

Deals
The item
The sale
When offer ends
Scroll
Signs-in
Denied
Accepted
History
Date
Info
Log-in Screen
Email
Password
Sign-up
Forget Password
User
Opens App
Menu
Deals
History
Drinks
Points
Settings
Sign out
Drink Menu
Drink Items
Index Page
Menu Button
Welcome message
Featured Menu Items
Menu
Success
Sign-up
-Contain Menu Items-
Points Page
Your Points
Featured redeemables
Redeemable item
The item
The reward
Redeem Button
Sign-up page
Email
Password
Name
Etc
Denied
Setting
Email
New Password option
Name
Telephone #
Etc.
Sign-out screen
Message

iii. Use Case Diagram

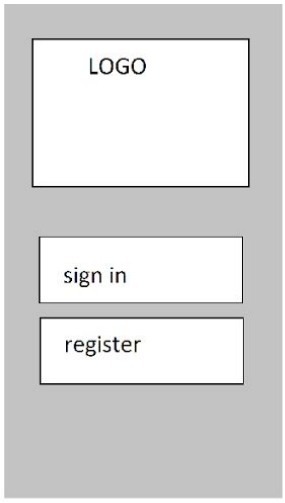



Web Server
initiates
IncorrectPassword
extends
ForgotPassword
initiates
AddressNotSaved
initiates
NoWebServerConnection
initiates
AccountAlreadyExists
initiates
NoCurrentDeals
initiates
UpdateBeacon
communicates
Beacon

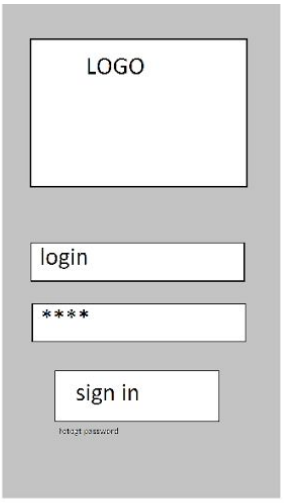
iv. Storyboard



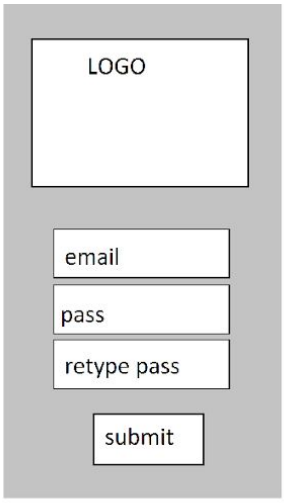
Splash Screen



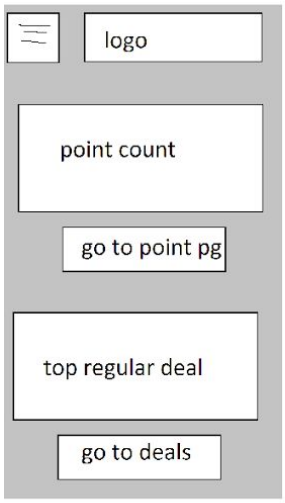
Login/Register Screen



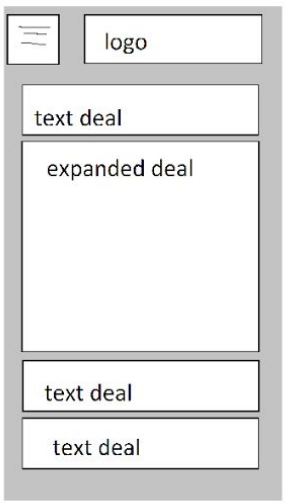
Login Screen



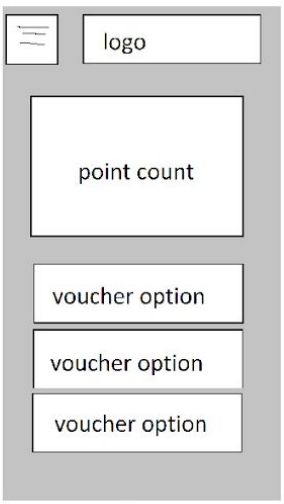
Register Screen



Frontpage Screen

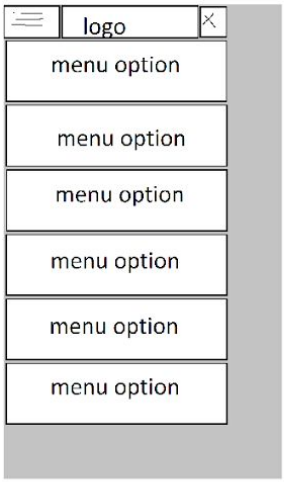


Deal Screen

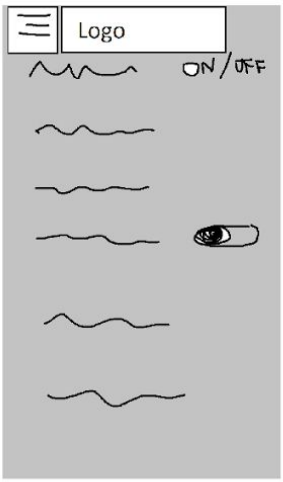


Point Voucher Screen

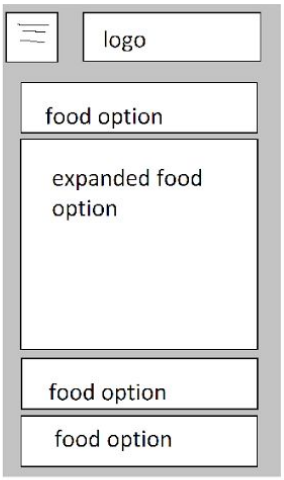
Welcome Screen



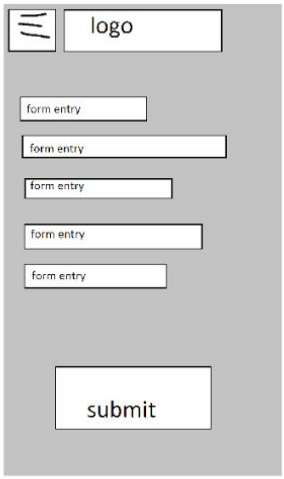
Menu Screen



Settings Screen



Food Menu Screen



Script Order Screen