CSC 431

UHungry?

Software Requirements Specification (SRS)

Team 10

Matthew Maya	Scrum Master and Software Developer
David Mills	Software Developer
Nicholas Sosnivka	Software Developer

Version History

Version	Date	Author(s)	Change Comments
1.1	02/20/2022	Matthew Maya David Mills Nicholas Sosnivka	• First draft

Table of Contents

Table of Tables

Table of Figures

131. System Requirements

131.1 Functional Requirements

1.1.1 Account Sign Up

Title	Sign Up
Description	App screen containing a form for users to register a new account
Priority	0
Precondition(s)	Application needs to be downloaded and a user without a valid login session must open the app
Basic Flow	User enters a valid email, username, password, and password confirmation in form boxes User presses sign up button which validates their form information, creates an account, and saves their data into a database
Postconditions(s)	If the form information is valid and an account is created successfully, a sign in session is created for that user and they are pushed to the school select screen If there is an issue with the form validation or account creation, the screen is updated and the user is prompted with an error message
Use Case Diagram	3.1

1.1.2 Account Sign In

Title	Sign In
Description	App screen to deal with signing in for users that are already registered
Priority	0
Precondition(s)	User must already have registered for an account and select the sign in button on the sign up screen.
Basic Flow	User enters their username and password in the respective text fields, User selects the “sign in” button, and a check is run to see if the user with those credentials exists in the database. User can also select the “reset password” button to update their password
Postconditions(s)	If a user with those credentials is found, a sign in session is created for that user and they are pushed to the university select screen If a user with those credentials is not found, the screen is updated and the user is prompted with an error message
Use Case Diagram	3.1

1.1.3 University Select Screen

Title	University Select
Description	App screen asking the user to select which university they are located at
Priority	1
Precondition(s)	A registered user with a valid log in session must open the app.
Basic Flow	User selects a drop down menu titled “Select University” User searches and selects their University from the drop down menu User selects “View University Map”, and the university data is saved in the database as an attribute for that user
Postconditions(s)	The user is pushed to the map page showing a map of that university
Use Case Diagram	3.1

1.1.4 University Map Screen

Title	University Map
Description	Map screen showing the university with the users location and all of the possible dining places on campus
Priority	1
Precondition(s)	The user has selected a University and has clicked the “Map” icon at the bottom of the screen
Basic Flow	User can see their location on the map represented by a circle icon, and bubble icons representing different dining spots on that University Campus User can zoom in and out of the map and drag the screen around view various areas, tapping a button in the top right corner will center the map view back around the user's current location User can select dining locations on the map by pressing a locations respective icon
Postconditions(s)	If a dining location is selected, a popup card appears detailing information about that location If user selects a different icon on the bottom of the screen, they are pushed to that screen
Use Case Diagram	3.1

1.1.5 Food Location Screen

Title	Food Locations
Description	A user must be able to view a list of the locations where they can get food on campus.
Priority	1

Precondition(s)	Users should have signed in to the application and clicked the "Food Locations" icon on the bottom of the screen to be redirected.
Basic Flow	User will see a list of food options relating to their selected university/campus The list will change depending on the users choice
Postconditions(s)	Selecting a food location will redirect the user to its respective dining location card on the map screen Selecting any other icon on the bottom of the screen will redirect the user to that screen (Map screen or Profile Details screen)
Use Case Diagram	3.1

1.1.6 Food Locations Filter Tab

Title	Sort By
Description	A user will be able to sort the collection of food locations based on various filtering options.
Priority	2
Precondition(s)	Users should have signed in to the application and clicked the "Food Locations" icon on the bottom of the screen to be redirected.
Basic Flow	User clicks the button in the top right of the screen where they can then select from various filtering options User will select filtering option depending on their mood altering the list of locations to eat at according to their choice
Postconditions(s)	if a filter option is selected then the list screen is updated
Use Case Diagram	3.1

1.1.7 Dining Location Cards

Title	Dining Location Cards
Description	Pop-up cards for each dining location presenting information about that location to the user
Priority	0
Precondition(s)	A user selects a dining location either from the university map screen or the food locations list screen.
Basic Flow	The card contains the name of the dining location, an image of the location, a brief description of the location, and the estimated waiting time for that location User can select a button "post waiting time" to update the estimated waiting time for the dining location.
Postconditions(s)	If the user selects "post waiting times" they are prompted with a form to input the estimated waiting time If the user selects anywhere on the screen outside of the card, the card will be minimized revealing the map screen behind it
Use Case Diagram	3.1

1.1.8 Post Waiting Time

Title	Post Waiting Time
Description	A form for users to post the waiting time at a dining location
Priority	0
Precondition(s)	A user must select the "Post Waiting Time" button while viewing a dining location card
Basic Flow	User is prompted with a form asking them to "Estimate the waiting time at this dining location" Users may select from a handful of options (<5 minutes, 5-10 minutes, 10-15 minutes, > 15 minutes), or input their own waiting time in minutes User selects the "Add your wait time input" button, and their data is saved in the database for that location and the estimated waiting time is updated averaging in the users input.
Postconditions(s)	Wait time is updated
Use Case Diagram	3.1

1.1.9 Search

Title	Search Tab
Description	A user must be able to look for the food location of their choosing.
Priority	2
Precondition(s)	Sign into application and either navigate to the Food Locations list or University Map screen.
Basic Flow	Users click on the search bar. Enter a few words based on the location they are trying to eat at. Application provides results based on what users enter.
Postconditions(s)	Locations saved in search are recorded by database, search history is also saved by database
Use Case Diagram	3.1

1.1.10 Profile Details

Title	Profile Screen
Description	A screen showing the users profile information
Priority	1
Precondition(s)	User selects the profile icon on the bottom of the screen
Basic Flow	User can view their basic profile information (email, username, current university) User can update their username, profile picture, or email by selecting "update profile", or update their university by selecting "update university" Users can press the "reset password" button to update their password.

Postconditions(s)	If another icon at the bottom of the screen is selected, the user is pushed to that page If "update profile" is selected a modal popup appears on the screen allowing the user to update any information If "update university" is selected, the user is pushed to the university select screen.
Use Case Diagram	3.1

1.1.11 Update Profile

Title	Update Profile Information
Description	A popup allowing the user to update their username, email, or profile picture
Priority	3
Precondition(s)	A user selects "update profile information" on the profile screen
Basic Flow	User is prompted with text fields for each piece of updatable information User may edit these texts fields to update whatever information they need to User selects "save" button, validating the updated information and saving it into the database
Postconditions(s)	If the new profile data is validated and saved into the database, the user is brought back to the profile screen If the data is not validated or saved, the popup is updated and the user is prompted with an error message.
Use Case Diagram	3.1

1.1.12 Password Recovery

Title	Password Recovery
Description	A user must be able to recover their password, should they forget it or their account is compromised.
Priority	0
Precondition(s)	User must have an account
Basic Flow	User has forgotten password, or account has been locked for security reasons User submits a request for a password reset to their email User receives email with password reset instructions User is redirected to an in-app password reset tool from unique list
Postconditions(s)	User password is modified, password is updated in database
Use Case Diagram	3.1

131.2 Non-Functional Requirements

1.2.1 Uptime

Title	Application Uptime

Description	The application must not be down, excluding any updates or maintenance changes
Priority	0
Applicable FR(s)	

1.2.2 Search Result Time

Title	Search Result Time
Description	Search result time must not take longer then 800ms
Priority	0
Applicable FR(s)	

1.2.3 Reset Password Server

Title	Reset Password Server
Description	Must have a server running at all times in order to send out reset password emails.
Priority	0
Applicable FR(s)	1.1.12

1.2.4 Data rendering time

Title	Data Rendering Time
Description	Any data that is fetched from the database is rendered onto the screen in less than 500ms
Priority	0
Applicable FR(s)	1.1.4, 1.15, 1.17, 1.1.9, 1.1.10

1.2.5 User Encryption

Title	User Data Encryption
Description	Any personal user data saved in the database must be encrypted to prevent that data from being stolen
Priority	0
Applicable FR(s)	1.1.1, 1.1.4, 1.1.11, 1.1.12

132. System Constraints

132.1 Tool Constraints

2.1.1 Flutter

Title	Flutter
Description	We will be using the Flutter app framework to develop the front end of our application
Priority	0

132.2 Language Constraints

2.2.1 Dart

Title	Dart
Description	The app will be programmed using the flutter IDE, so knowledge of dart will be required
Priority	0

2.2.2 PHP

Title	PHP
Description	The backend framework and api requests will be written in PHP
Priority	0

2.2.3 MySQL

Title	SQL
Description	The data for the users, universities, and dining locations will be saved in a MySQL database, so the SQL language must be used
Priority	0

132.3 Platform Constraints

2.3.1 IOS or Android

Title	IOS or Android
Description	Flutter allows for cross platform deployment, allowing the app to be created for both ios and android simultaneously
Priority	0

132.4 Hardware Constraints

2.4.1 Mobile Device

Title	Mobile Device
Description	The app would be available to anybody who has a mobile device running on IOS or Android software
Priority	0

132.5 Network Constraints

2.5.1 Database Access

Title	Database Access
Description	The database must be hosted on a network so it can be accessed to save or query data at all times when the app is running
Priority	0

132.6 Deployment Constraints

2.6.1 Amazon Web Services

Title	Amazon Web Services
Description	the database for the application will be deployed on Amazon Web Services using the Amazon RDS package for the MySQL database
Priority	0

132.7 Transition & Support Constraints

2.7.1 Github Version Control

Title	Github Version Control
Description	We will be using Github Repository as version control for the code. Since this project must be completed by the end of the semester, We will have to implement the application at a later time.
Priority	0

132.8 Budget & Schedule Constraints

2.8.1 Time Constraints

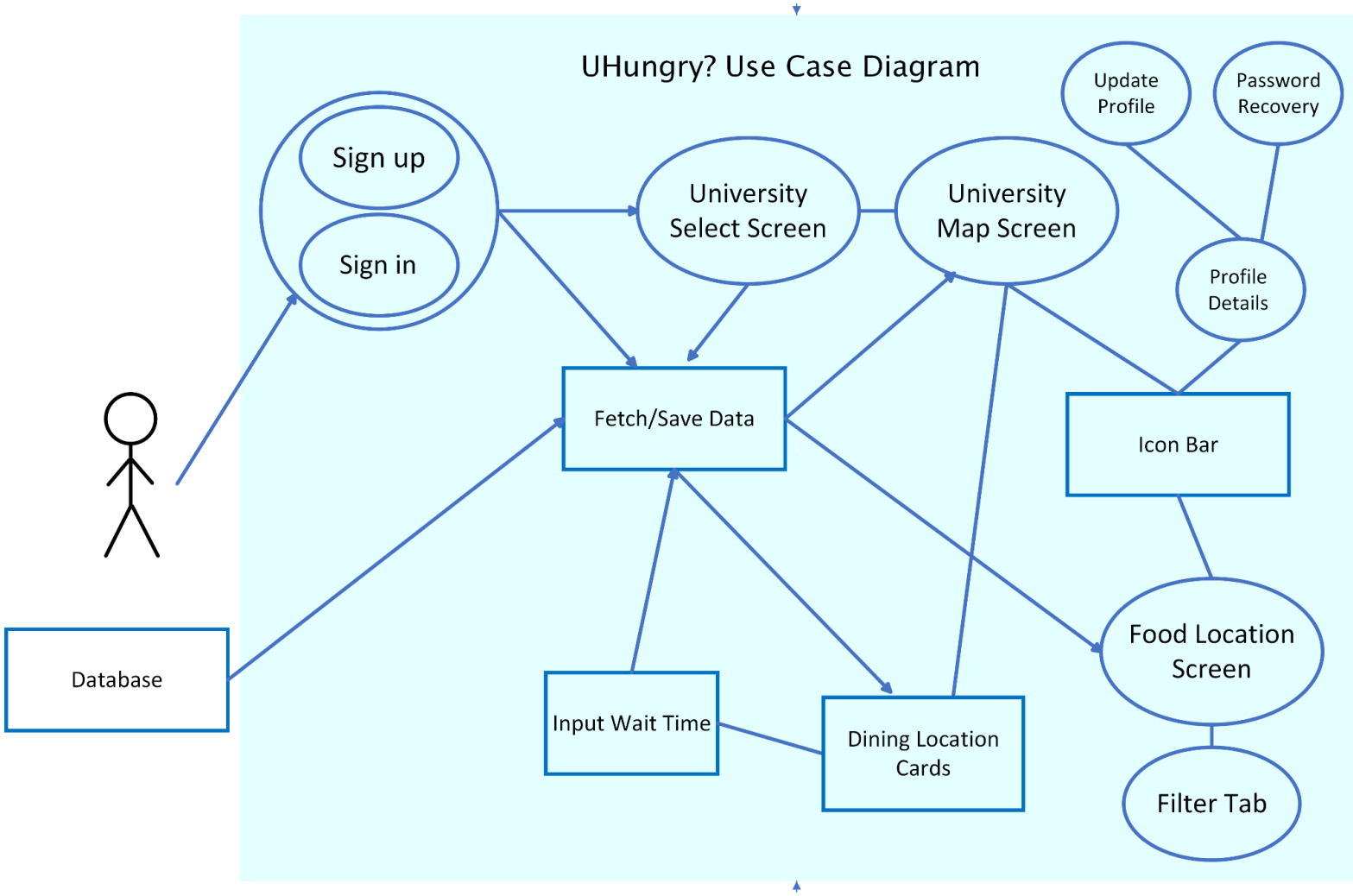
Title	Time Constraints
Description	The mockup for the project must be completed by the end of the semester (May 2022)
Priority	0

2.8.2 Budget Constraints

Title	Budget Constraints
Description	There are no funds available for this project, therefore there is no budget
Priority	0

133. Requirements Modeling

3.1.1 Use Case Diagram



134. Evolutionary Requirements

134.1 Functional Requirements

4.1.1 Supported Universities

Title	Supported Universities
Description	Eventually, more universities should be supported on the application that way the application can become more widespread increasing the user base
Priority	4
Precondition(s)	The app must get a large enough amount of users to justify adding univerisities Data for the new universities and their dining spots need to be added to the database
Postconditions(s)	The university will be added as an option on the university select screen
Use Case Diagram	

4.1.2 New Food Location Option

Title	New Food Locations Option
Description	Over time universities are constantly changing and improving their food options in order to meet the students needs which is why the option for users to suggest that a new food location will be supported on the application.
Priority	4
Precondition(s)	- The University must bring new food options onto campus in order for the spot to be registered as a location to get food. - The location must get enough requests from users and a background check must be done on the location before adding it to the list.
Postconditions(s)	The new location will be added as an option on the map and list screen as well as in the database with all the particular information relating to that food option.
Use Case Diagram	

134.2 Non-Functional Requirements

4.2.1 Upgrade Allocated Servers

Title	Upgrade Allocated Servers
Description	As the user base scales, more servers and databases will be required to hold all of the user data and access it quickly
Priority	3
Applicable FR(s)	4.1.1, 4.1.2