Raincheck

Project Team: Lauren Hall, Marissa Martin, Billy Marion, and Parth Patel

# Table of Contents

[**Table of Contents**](#_m8xx9jyuwenv) **1**

[**Project Definition**](#_q08w2y8mtgdx) **2**

[Environment and System Models](#_hvgzs4z17d11) 2

[**Project Requirements**](#_xozphasppv1w) **2**

[**Functional Requirements**](#_2nx1eaqvjx15) **2**

[**Quality Requirements**](#_cgq4qi98tqlg) **3**

[**Platform Requirements**](#_7ksp7fpa5ycw) **3**

[**Process Requirements**](#_jvo399l6ncro) **3**

[**Use-Cases**](#_kvrfznnmnpoe) **3**

[Use-Case Diagram](#_1maf4sn5kxfz) **4**

[**Domain Analysis**](#_yc6ofac4np2i) **4**

[Glossary](#_s7gs7bfj7xbv) **4**

[Extensions](#_c2450onihyek) **5**

[General Knowledge about the Domain](#_xw38k9m9qbmf) **5**

[Clients and Users](#_ks01jyom61rw) **5**

[The Environment](#_u0fhkatistpx) **6**

[Tasks and Procedures](#_5qowmy67nqpw) **6**

[Competing Software](#_m0cld3r446a5) **6**

[**Project Specification**](#_tcy50k8f4kg3) **7**

[UML Diagrams](#_tc1jwvidyo82) **7**

[System Diagram](#_4lbq4a59un6j) 7

[Use-Case Diagram](#_coype1own0t) 7

[State Diagram](#_r830h89xetgh) 8

[Activity Diagram](#_xpfmburbon9c) 9

[GUI Diagrams](#_lh75fpw9p4wy) **10**

[Home Screen](#_3t9jiegn89db) 10

[User Settings](#_fj1kfbgwrllf) 11

[Location](#_mrdsrou4pysd) 12

[**Subsystem**](#_u2m8xsyb49u) **13**

# Project Definition

Our project is named Raincheck. It is neither strictly greenfield nor a derivative project, as it is a new application that runs using pre-existing data. The software is generic, and its accessible to anyone who has internet access, basic reading skills, and a cellular device that runs on Android software. The main user group would consist of a variety of ages and demographics, but would focus on people who want to receive notifications about upcoming weather.

Problem Statement: To provide an application that alerts its users to upcoming inclement weather and suggestions on how they should prepare for it, in an unobtrusive and convenient format that doesn’t require the user to otherwise have to check the weather report when in a hurry.

## Environment and System Models

Raincheck will run on any hardware that supports Android software 6.0 or higher and maintains an internet connection. Raincheck is an application and not accessible through web browser systems.

# Project Requirements

## Functional Requirements

1. The system will provide customized weather forecast notifications based on user-provided settings.
2. The system will have three primary interface windows:
   1. The login page, where the user will be able to create a login in order to save their data or sign into their previously created account.
   2. The home page, where primary weather information is displayed.
   3. The settings page, where the user can customize their personal settings.
3. When the application is opened, the user will need to login or register in order to access to the software.
4. Users will be allowed to update and change various settings within the application, such as the times notifications are sent, what types of weather to warn for, the location, and the temperature format that results are displayed in.

## Quality Requirements

1. In situations without Internet access, the system will utilize error-handling to ensure that incorrect data is not sent to the user.
2. The system itself will be constructed in such a way that future updates and features can be added.
3. The system will prevent users from inputting invalid settings.
4. In case of system failure or corruption, user data will be saved in a cloud-based database, so that users will have their data saved if redownloading the application on a new device.
5. The system will fetch data in the background at time intervals designated by the user’s preference
6. The system will operate quickly and efficiently, without using large amounts of data.

## Platform Requirements

1. Computing Platform: As an Android application, the application will be able to run on any system that supports Android 6.0 or higher.
2. Technology Used: The application will be written using Java in Android Studio.

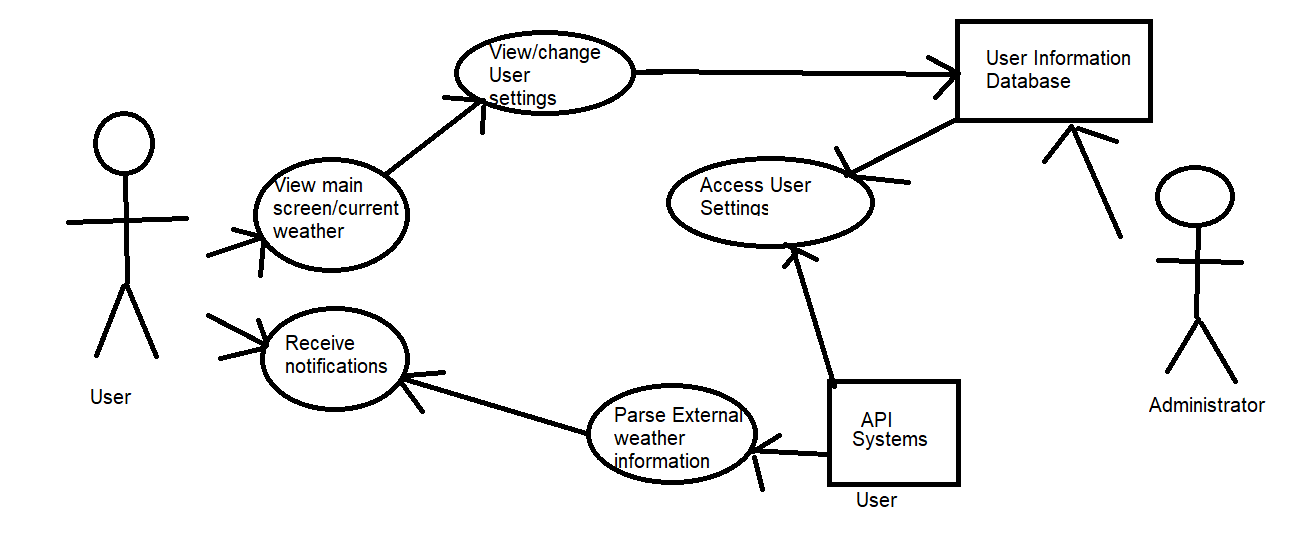
## Process Requirements

The project must be completed by April 28th, 2019 with the requirements document due February 10th, the specification document due March 12th, and the source code due March 28th.

## Use-Cases

1. Use-case: User Registration
   1. Actors: New Users
   2. Summary: When a new user wants to create an account, they must provide registration information.
   3. Steps:
      1. Click on “New User”
      2. Create username and password
2. Use-case: User Login
   1. Actors: Returning Users
   2. Summary: When a returning user wants to access their account, they must enter their login information.
   3. Steps:
      1. Click on “Returning User”
      2. Enter username and password
3. Use-case: Change Settings
   1. Actors: Users
   2. Summary: A user may access their settings and have the option to change them.
   3. Steps:
      1. Click on “Settings”
      2. Click desired setting to change.
      3. Edit settings using application interface.
4. Use-case: Check Weather
   1. Actors: Users
   2. Summary: A user may view the weather for any chosen location.
   3. Steps:
      1. Open the application; weather information is located on the front page.
5. Use-case: Send Notification to User
   1. Actors: Users
   2. Summary: A user will receive an alert on their phone based on their user settings.
   3. Steps:
      1. Set time for notification using the user settings interface.
      2. Wait for the notification, which will appear at the designated time.

### Use-Case Diagram



## Domain Analysis

### GUI Diagram

### Glossary

1. **Administrator**: A non-user entity responsible for maintaining and streamlining the data flow between API systems, Database system, and the User.
2. **Inclement Weather**: Harsh weather, typically cold and wet. In the specific circumstances of Raincheck, inclement weather is typically rainy, cold, or requires preparation on behalf of the application user.
3. **Notification**: A text banner that appears at a designated time on the lock screen of the cellular device the application is installed on.
4. **F/C:** Fahrenheit and Celsius temperature measures.

### Extensions

Possible future extensions of this application will include the implementation of cross-platform functionality to be able to run on other operating systems including iOS. Other functionalities that may be included in future extensions are expanding the types of weather conditions that user may check for, such as snow, high winds, and humidity, and the tracking of weather conditions between two points of travel for a user, so that the user may be forewarned of weather conditions that they may encounter on their travels from point A to point B. If found helpful by potential users, a ‘help’ page could be implemented that gives more information about what each user setting did.

### General Knowledge about the Domain

1. People who check the weather in the morning are often leaving their house during the day.
   1. These people may go to several different locations other than the location where weather information is being gathered.
2. People who check the weather in the morning usually will be outside at some point during the day.
3. People who would rather have a notification about what to bring for the day rather than check the forecast themselves may be in a hurry, meaning they might not have a lot of time to get ready.

#### Clients and Users

1. Potential customers would be any with a desire to have a clear method of obtaining weather information regarding possible inclement weather.
2. Individual user may include:
   1. Customer, who would make use of the application’s primary functionality, which is to provide customized weather forecast notifications based on user-provided settings.
   2. System Administrators, who would manage both the application’s database and the application environment.

#### The Environment

1. All actors will have access to the internet, and a device that uses both location services and an Android operating system. The application will be able to operate on any device running Android 6.0 or above.

#### Tasks and Procedures

Current methods for finding information regarding possible inclement weather can be generalized as follows:

1. Watching the News: A person watches a news channel until that channel broadcasts their weather report.
   1. This approach will often involve a person waiting until specific times in order to watch a news station's weather report.
   2. Can be very inconvenient when a person is busy or planning on travelling, as it requires that a person be in front of a television and on a specific channel at a particular time, along with the news station only covering the forecast for nearby locations.
2. An Internet Search: A person uses a web browser in order to search for weather information in their desired location.
   1. This approach allows a person to obtain weather information at any given moment for a location, but requires that they open a web browser and search for the desired information.
3. Viewing an existing weather application: A person downloads an existing weather app onto their mobile device.
   1. This approach allows a person to obtain weather information at any given moment for a saved location, but requires that they open the application in most cases and may provide more information than desired, which forces them to search the screen for the desired information.

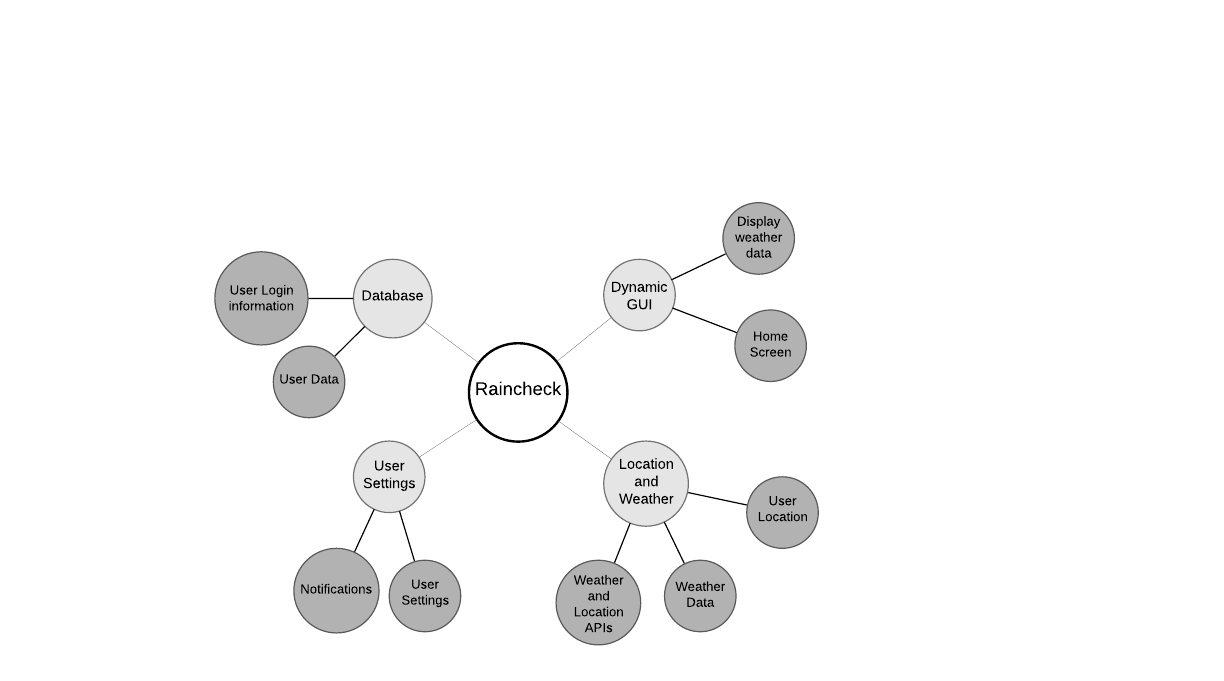
#### Competing Software

1. There are multiple other applications that tackle the same problem of customizable weather alerts, which are unoptimized to our specific user demographic. Some examples follow:
   1. Rain Alarm for Google Play: Also provides users with upcoming rain forecasts specialized to their location, but does so using multiple informationally dense pages, includes disruptive ads, and lets users know the rain forecast by how far away it is from their location rather than sending them an alert.
   2. Will it Rain and Will it Snow for the Apple App Store: Both of these applications send users notifications at a specified time about inclement weather, but are prohibitively specific in their functionality (only rain alerts or only snow alerts), and are not available for Android users.

# Project Specification

## UML Diagrams

### System Diagram



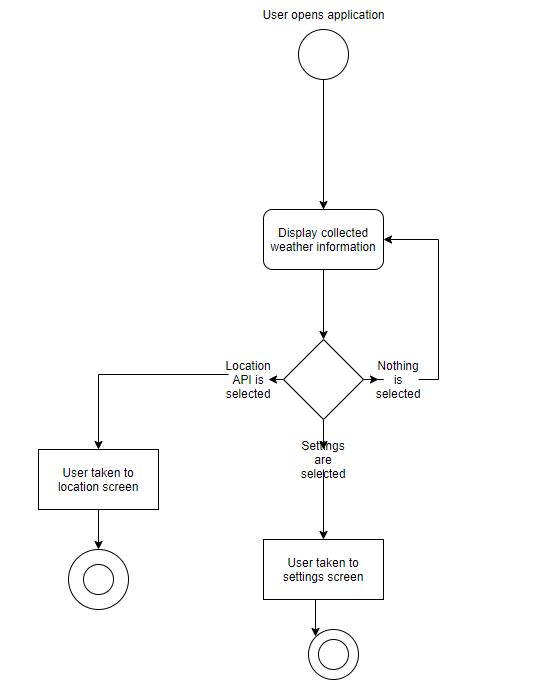
The Raincheck application is split up into four subsystems that handle various types of information collecting, data storage, data parsing, and user interaction handling. The system division has been constructed in such a way that the entire system requires each subsystem in order to properly function, and many subsystems have cross-interactions that allow them to effectively utilize the collected data.

### Use-Case Diagram

*Please reference the Use-Case Diagram subheading on pg. 4 of this document.*

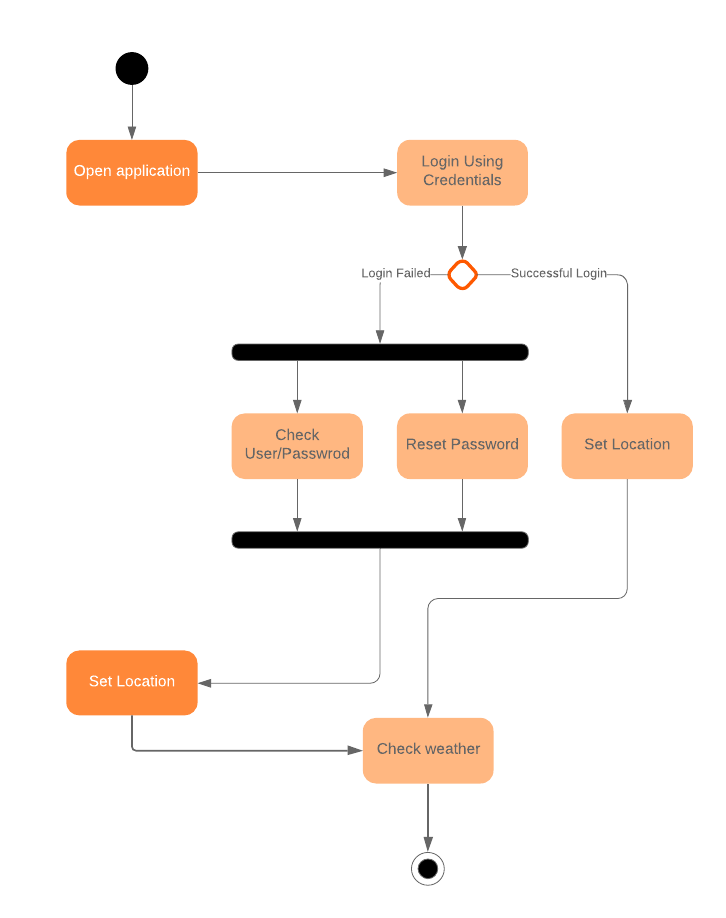
### 

### State Diagram



The main screen of Raincheck is a dynamic GUI upon which the view changes according to what weather information is currently connected; apart from this, the main processes that happen in the root menu are either transporting the user to a different screen in which they may add information to their account that allows the application to provide a more specific weather information. If the user does not select either of these two options, the screen continues displaying the current weather until the application is closed.

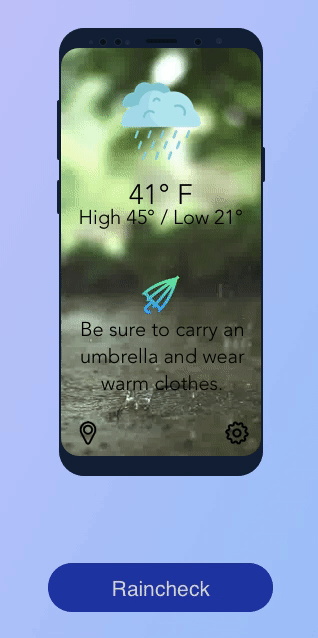
### Activity Diagram



The main screen of the application is a dynamic GUI that displays the current weather information and possesses a dynamic background. The activities involved are asking the user for login credentials upon starting the app for the first time and the application goes through normal login procedure and asks for location for which the user would like to know weather and the application displays dynamic weather information.

## GUI Diagrams

### Home Screen



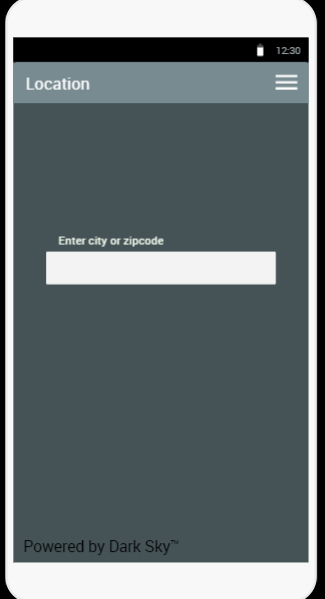
This is an example of the home page of the application which provides the user with accurate weather information and it also has two buttons for settings and location which can be touched to navigate to different sections of the application for different functionalities.

### User Settings

### 

This is an example of the settings screen that allows users to alter their preferences on what information they want to be sent to their device and at what time it occurs. It also allows users to log in to their account, handled by the user database system.

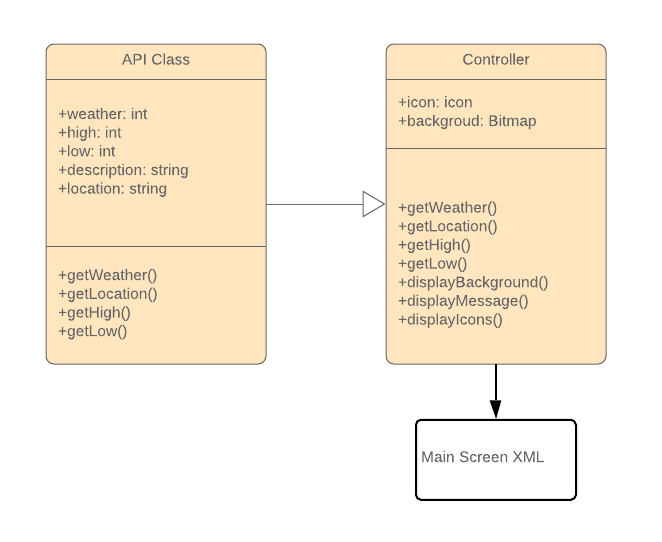
### Location



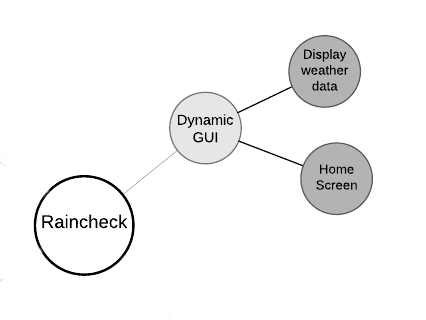
This is an example of the locations screen that allows users to input the location that they want their application to show weather information for. It also allows users to backtrack to the home page, if necessary.

# Subsystem

1. Class Diagram



1. Data dictionary
   1. There will be a collection folder consisting of different dynamic animations which the controller has access to.
2. Since the subsystem is the Graphics Handler and it controls the main screen of the application, it will have the same GUI as the main screen. *Please refer to the GUI on page. 10*
3. System Diagram



1. *Please refer to GUI document for details on GUI design.*