**CMPE 131**

**Software Engineering**

**Moo Project**

Detailed Design Specification

Paul Diaz

Stefan Francisco

Timothy Magallanes

Tyler Olson

The following Detailed Design Document illustrates diagrams, classes, tables and architectural views of the mobile application Moo in accordance with (IAW) the team’s detailed Software Requirements Specification (SRS). The entirety of this document addresses the mobile application’s requirements emphasizing the principal goals of software design; sufficiency, understandability, modularity, high cohesion, low coupling, robustness, flexibility, reusability, information hiding, efficiency, and reliability.

This document serves as the primary source of blueprints for the mobile application Moo. The designs detailed below outline the communication between the mobile application Moo and other system components such as Firebase and Rotten Tomatoes’ API. Additionally, this shows how the application will be structured during the implementation phase.

The mobile application Moo will be structured and modeled using Model View Controller architecture (MVC), Strategy Pattern, and Delegation Pattern.

**Table of Contents**

[Context Model](#_7sz6z3jphgi4) ………………………………………………………………………………………………. 4

[MVC Architecture](#_4z1g0235fh2a) ………………………………………………………………………………………………. 5

[Process Model](#_o3wla6pnq3z3) ………………………………………………………………………………………………...6

[State Transition Diagram](#_twicr9idjfkd) …………………………………………………………………………………7

[Use Cases](#_k3cn7oc3zhb5)

[Moo App Installation](#_8ivmgxeig3zw) …………………………………………………………………………………..8

[Use Case: Registration](#_osj8geebcel7) …………………………………………………………………………………10

[Use Case: Login](#_v49t81nw9qcz) …………………………………………………………………………………………...11

[Use Case: Browse Movies](#_up0if684v6yl) …………………………………………………………………………….12

[Use Case: View and/or Edit Profile](#_mw6a9cppov4c) ……………………………………………………………...13

[Use Case: Critique a Movie](#_mrv1gxsx3ej0) ………………………………………………………………………….14

[Use Case: Critique a Movie Trailer](#_cyczelnbfqks) ……………………………………………………………….15

[Use Case: Leaderboard](#_jjtclrv6wjwz) ……………………………………………………………………………….17

[Use Case: Redeem Points](#_5jha561mao4q) ……………………………………………………………………………18

[Sequence Diagrams](#_f4q9gqiwjiui)

Installation…………………………………………………………………………………………………21

[Login](#_ju5t6mxlti6x) ………………………………………………………………………………………………………...22

[Login Variation 2](#_e2aigx567a86) ……………………………………………………………………………………….23

[Registration](#_ocptnv9nzx6o) ……………………………………………………………………………………………...24

[Registration Variation 2](#_met0133r9jc1) …………………………………………………………………………….25

[Browse Movies](#_38hbqpeyzpld) ………………………………………………………………………………………....26

View or/and Edit Profile …………………………………………………………………………...27

Critique a Movie ……………………………………………………………………………………….28

Critique a Movie Trailer …………………………………………………………………………….28

Leaderboard …………………………………………………………………………………………….29

Redeem Points ………………………………………………………………………………………...30

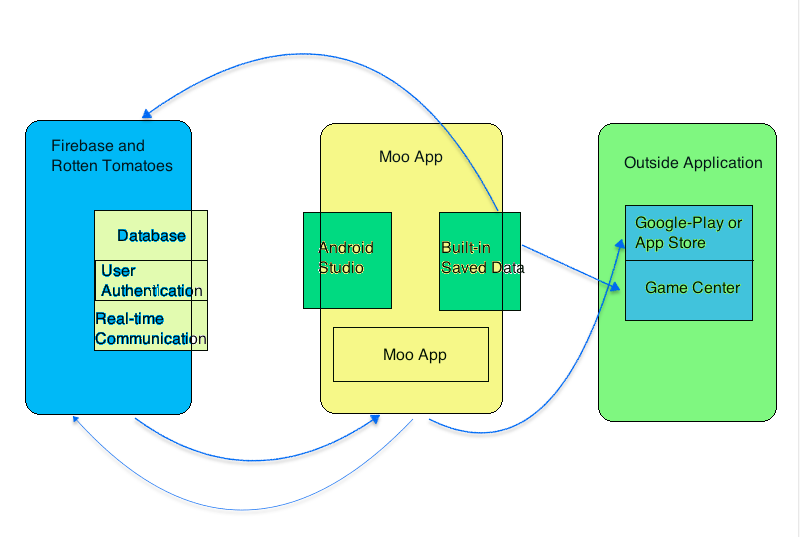
[Data Flow Diagrams](#_987z4rdv2n5o)  ……………………………………………………………………………………...31

[Class Diagram](#_qff19syeyn8n) …………………………………………………………………………………………….....33

[Architectural View of Deployment](#_1y52ykm5urqe)  ………………………………………………………………...35

[Traceability Matrix](#_qofpede0bhd9) ………………………………………………………………………………………..36

# Context Model

The context model represents context of Moo system and how it will connect to the external entities, other outside application.

# MVC Architecture

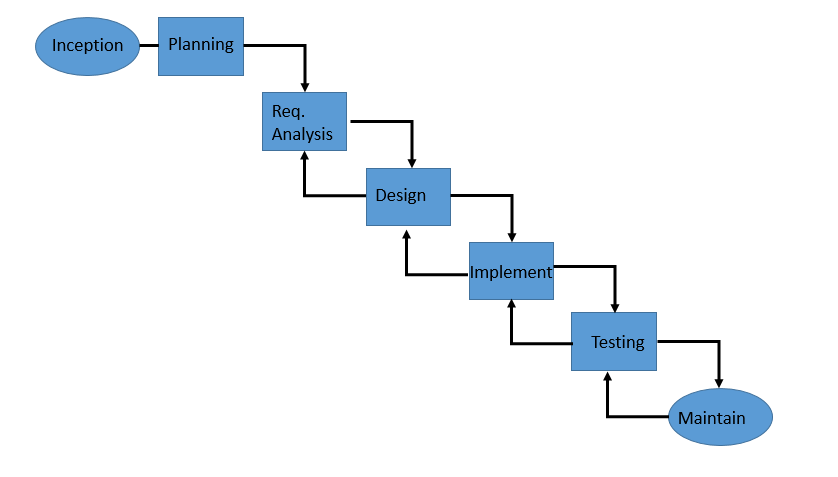
# The diagram below shows the mobile application Moo’s software design pattern architecture. Each of the three sections are responsible for their own particular portion. Model, the lowest level of the design pattern, is responsible for all data relating to the Moo application, including user data and movie data. Next, view is is for displaying and showcasing all or a portion of the data to the user, such as comments, movie reviews, trailers, incentives, and more. Lastly, the controller portion denotes the software code for controlling the interaction between view and the model.

# 

# 

# 

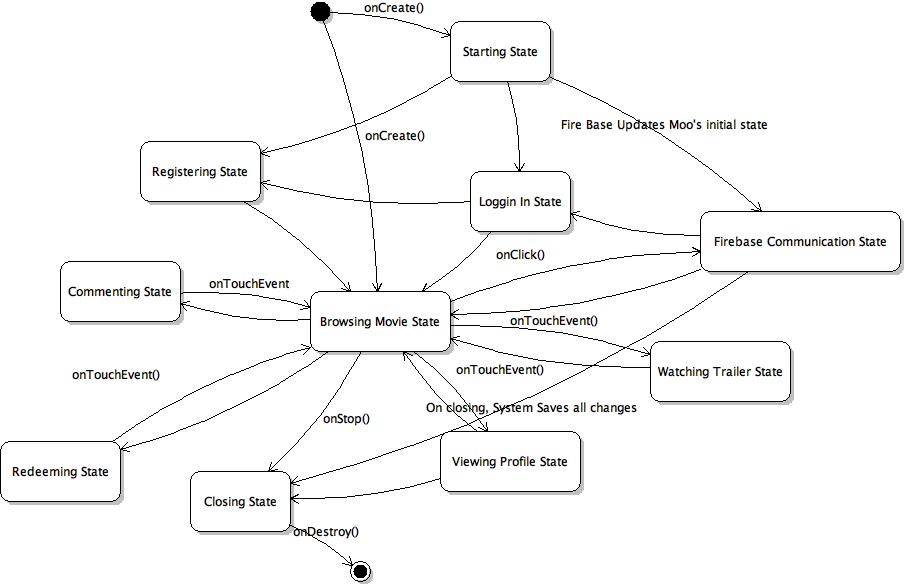
# Process Model

****

The team will be using a waterfall process model, which is a sequential process meaning one phase cannot begin until the previous phase is complete. The Process model begins with the inception of the project. In this case, the team received a request for proposal (RFP) of a Moo project, thus beginning the inception phase. A mobile and/or web application was desired to be used for rating movies in an incentivized way. Following this, the team set up a gantt chart in the planning phase as an initial schedule for the project as well as determined costs for the project and additional necessary resources. The requirement analysis phase answers the simple question of “what will the application do?” The team accomplished this by following the Institute of Electrical and Electronics Engineers (IEEE) 830-1998 recommended practice for Software Requirement Specifications (SRS). This answered the question of what the Moo application will do. Next, the design phase consists of how the team ensures the application will accomplish what the RFP requested and its intended purpose for future stakeholders. This can be seen by looking at the software architecture. The implementation phase includes writing the code and getting a baseline working model. Testing includes checking to ensure the software and code written in the implementation phase works. Validation is used to ensure the software meets user requirements. Lastly, the team will maintain the software repairing defects, adding capability, and modify the code and software to adhere to user requests in the future. The arrow loops denoted on the process model indicate a loop occurring, which can happen for a number of reasons. If stakeholders deem the software too buggy or note too many defects, the process loops back to the testing phase where the team can perform more adequate testing.

# State Transition Diagram

The mobile application Moo is divided up into several states, which can be viewed below in the state transition diagram. These states denote phases of the mobile application for the team as developers as well as stakeholders as users. By dividing the application up into these several states, the team knows the application will always be in one specific state. Each state is defined based off of the team’s previous software requirement specifications (SRS).



# Use Cases

The use cases listed below denote major functions within the mobile application Moo. Each of the below sections describe relationships and interactions between users of the mobile application Moo and the system application itself, thus explaining how the application is utilized by stakeholders and users.

## Moo App Installation

|  |  |  |
| --- | --- | --- |
| Use Case: | Installation: |  |
| Steps | User Action | System Action |
| 1 | User go to App Store or Play Store |  |
| 2 |  | System Loads App Store or Play Store |
| 3 | User types "Moo" in the search engine |  |
| 4 |  | System sets Moo to be ready for download |
| 5 | User hits download button |  |
| 6 |  | System confirms user's authorization by providing Terms and Condition of the purchase |
| 7 | User hits accept button |  |
| 8 |  | System starts downloading Moo |
|  |  |  |
| Variation II | New App Store or Play Store user |  |
| 6 |  | System will ask the user to provide credit card information in order to purchase the application |
| 7 | User inputs Name, credit card information, and accepts terms and condition |  |
| 8 |  | System confirms credit card validation |
| 9 | User hits download button |  |
| 10 |  | System confirms user's authorization by providing Terms and Condition of the purchase |
| 11 | User hits accept button |  |
| 12 |  | System starts downloading Moo |

## Use Case: Registration

|  |  |  |
| --- | --- | --- |
| Use Case: | Registration: |  |
| Steps | User Action | System Action |
| 1 | User hits "Sign up" to register as a user |  |
| 2 |  | System provides a window and wait for user's input for personal information. |
| 3 | User inputs name, birthday, email, username, password, and alternative email address |  |
| 4 |  | System accepts information to create a file for the new user |
| 5 | User hits "Register" |  |
| 6 |  | System displays Moo's main menu |
|  |  |  |
| Variation II | Email Address Already Registered |  |
| 4 |  | System informs user that email address is already registered |
| 5 |  | System provides a window to recover password |
| 6 | User hits "recover password" |  |
| 7 |  | System sends confirmation to the alternative email address to retrieve temporary password |
| 8 | User go to alternative email to get the temporary password |  |
| 9 | User inputs temporary password |  |
| 10 |  | System authenticates the temporary password |
| 11 |  | System asks user for new password |
| 12 | User inputs new password twice |  |
| 13 |  | System accepts information and create a file for the new user |
|  |  |  |

## Use Case: Login

|  |  |  |
| --- | --- | --- |
| Use Case: | Login |  |
| Steps | User Action | System Action |
| 1 | User taps Moo App |  |
| 2 |  | System displays Login and ask for email and password |
| 3 | User types email and password |  |
| 4 |  | System connects to Firebase and confirms user’s email and password |
| 5 |  | System displays main menu where user can start browsing for movies |
| Variation II | Forgot password |  |
| 5 | User clicks on “Forgot Password” button |  |
| 6 |  | System asks for email, and sends a confirmation email to reset password |
| 7 | User confirms received email and enter new password |  |
| 8 |  | System connects to Firebase and updates registered password |
| 10 |  | System displays main menu where user can start browsing for movies |

## Use Case: Browse Movies

|  |  |  |
| --- | --- | --- |
| Use Case: | Browse movies |  |
| Precondition | Application is on main menu where list of movies are shown |  |
| Steps | User Action | System Action |
| 1 | User starts to swipe the screen, left or right, to browse movies |  |
| 2 |  | System shows the list of featured movies |
| 3 | User choose specific movie |  |
| 4 |  | System displays movie information; movie main image, movie plot, actors, director, rating, and critiques. |
| 5 | User chooses to watch trailer: see use case watch trailer |  |
| 6 | User chooses to critique the movie: see use case critique a movie |  |

## Use Case: View and/or Edit Profile

|  |  |  |
| --- | --- | --- |
| Use Case: | View and Edit Profile |  |
| Steps | User Action | System Action |
| 1 | User taps on profile icon to view user's profile |  |
| 2 |  | System opens a new window displaying user's information |
| 3 | User chooses to edit profile |  |
| 4 | User starts to taps on whichever field choose to edit |  |
| 5 | When user done editing, user taps "Save" button |  |
| 6 |  | System updates user's profile |

## Use Case: Critique a Movie

|  |  |  |
| --- | --- | --- |
| Use Case: | Critique a Movie |  |
| Steps | User Action | System Action |
| 1 | User taps on "Critique Movie" button |  |
| 2 |  | System expands text area and waits for user's critique |
| 3 | User writes critique |  |
| 4 |  | System accepts user's critique and saves it |
| 5 |  | System updates user's Moo points |
| Variation I | User Gives a Comment to Other User's Critique |  |
| Steps | User Action | System Action |
| 1 | User taps on "Reply" button |  |
| 2 |  | System expands text area and waits for user's comments |
| 3 | User writes comment |  |
| 4 |  | System accepts user's comment |
| 5 |  | System updates user's Moo points |
| Variation II | User likes someone else's critique or comment |  |
| Steps | User Action | System Action |
| 1 | User hits "like" to a specific critique or comment |  |
| 2 |  | System updates the number of likes of the said critique or comment |
| 3 |  | System updates user's Moo points |

## Use Case: Critique a Movie Trailer

|  |  |  |
| --- | --- | --- |
| Use Case: | Critique a Movie |  |
| Steps | User Action | System Action |
| 1 | User taps on "Critique Movie" button |  |
| 2 |  | System expands text area and waits for user's critique |
| 3 | User writes critique |  |
| 4 |  | System accepts user's critique and saves it |
| 5 |  | System updates user's Moo points |
| Variation I | User Gives a Comment to Other User's Critique |  |
| Steps | User Action | System Action |
| 1 | User taps on "Reply" button |  |
| 2 |  | System expands text area and waits for user's comments |
| 3 | User writes comment |  |
| 4 |  | System accepts user's comment |
| 5 |  | System updates user's Moo points |
| Variation II | User likes someone else's critique or comment |  |
| Steps | User Action | System Action |
| 1 | User hits "like" to a specific critique or comment |  |
| 2 |  | System updates the number of likes of the said critique or comment |
| 3 |  | System updates user's Moo points |

## Use Case: Leaderboard

|  |  |  |
| --- | --- | --- |
| Use Case: | Leaderboard |  |
| Steps | User Action | System Action |
| 1 | User taps on "view leaderboard" button |  |
| 2 |  | System asks user what type of leaderboard: movie or users |
| Variation I | User chooses movie leaderboard |  |
| Steps | User Action | System Action |
| 3 | User chooses movie leaderboard |  |
| 4 |  | System displays movie leaderboard |
| Variation II | User chooses user leaderboar |  |
| Steps | User Action | System Action |
| 3 | User chooses user leaderboard |  |
| 4 |  | System displays movie leaderboard |

## Use Case: Redeem Points

|  |  |  |
| --- | --- | --- |
| Use Case: | Redeem Points |  |
| Steps | User Action | System Action |
| 1 | User "Redeem Points" button |  |
| 2 |  | System asks user's redeem preference: store credits or monetary Rewards |
| Variation I | User chooses store credits |  |
| Steps | User Action | System Action |
| 3 | User chooses store credits |  |
| 4 |  | System issues an electronic stab (that contains a fund) via email or text |
| 5 | User receives the electronic stab |  |
| 6 |  | System updates user's Moo points by subtracting the corresponding amount of Store credits chosen by the user |
| Variation 2 | User chooses monetary rewards |  |
| Steps | User Action | System Action |
| 3 | User chooses monetary rewards |  |
| 4 |  | System issues an electronic stab (that contains a fund) via email or text |
| 5 | User receives the electronic stab |  |
| 6 |  | System updates user's Moo points by subtracting the corresponding amount of Store credits chosen by the user |

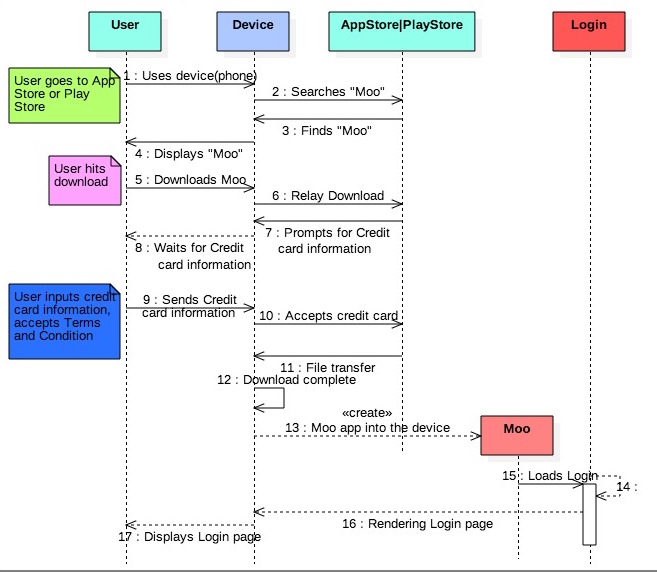
# 

# 

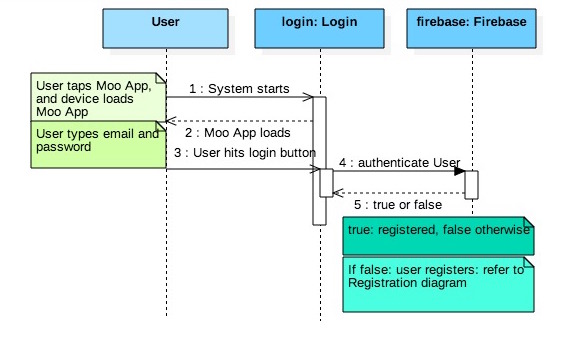
# Sequence Diagrams

The below sections showcase diagrams of objects within the mobile application Moo. Each diagram shows a sequence of functionality calls between each object. The sequence diagrams below show messages between objects within the software design.

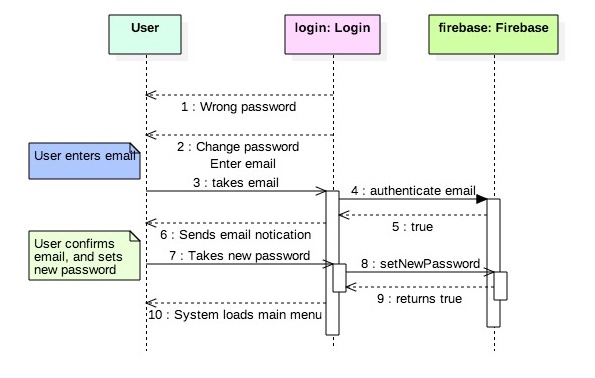
## Installation



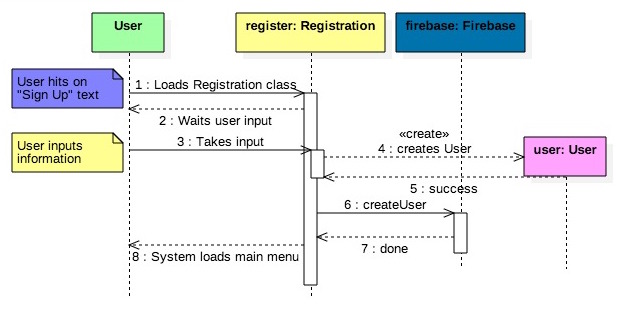
## Login Variation I



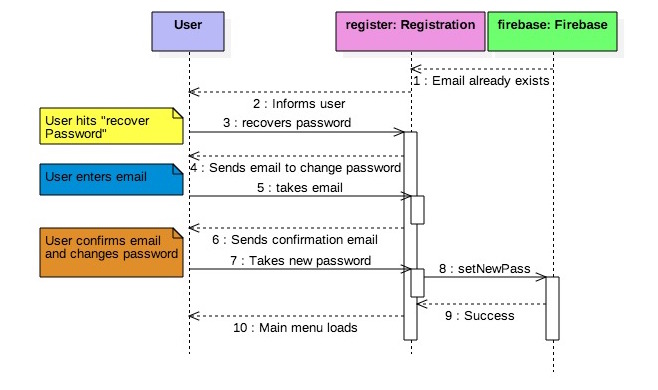
## Login Variation II



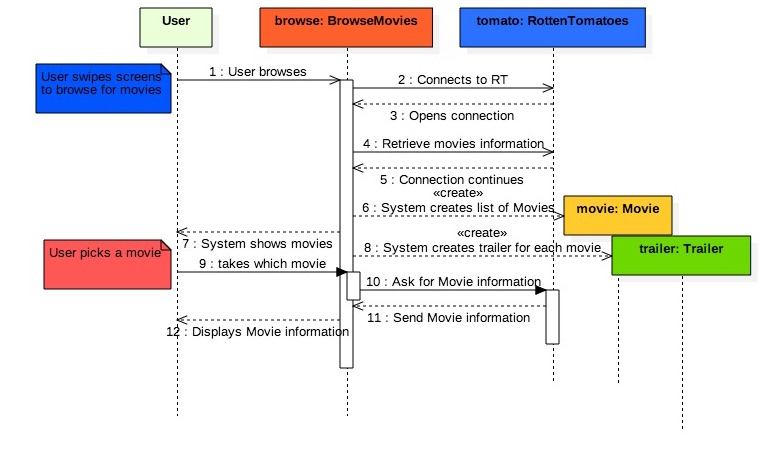
## Registration



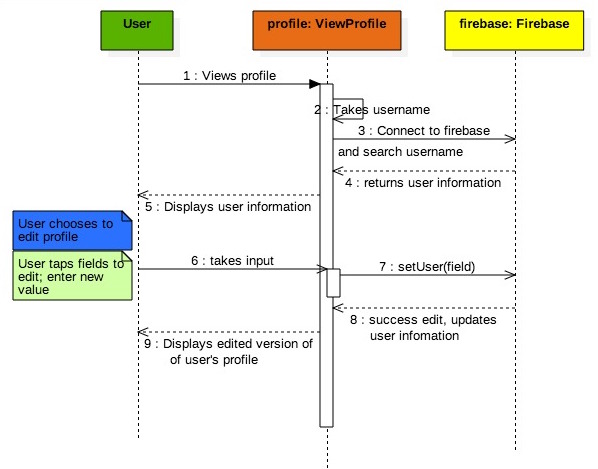
## Registration Variation 2



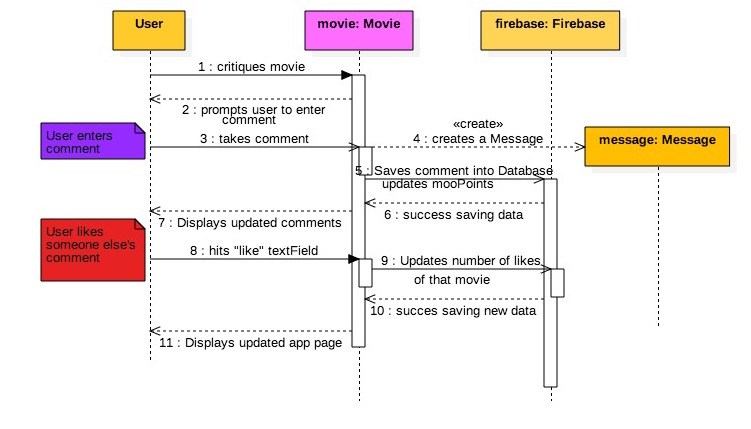
## Browse Movies



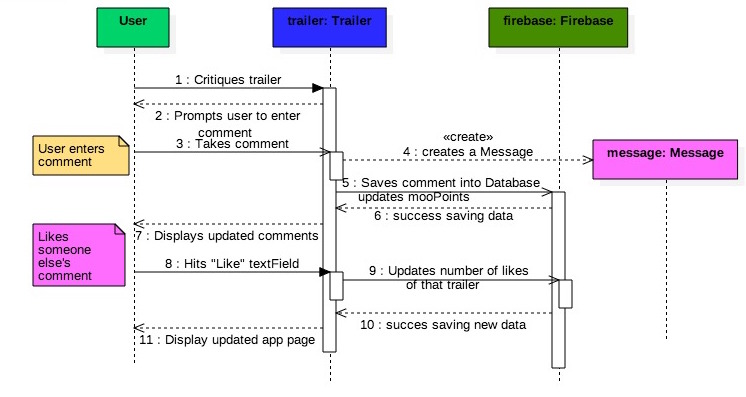
## View and/or Edit Profile



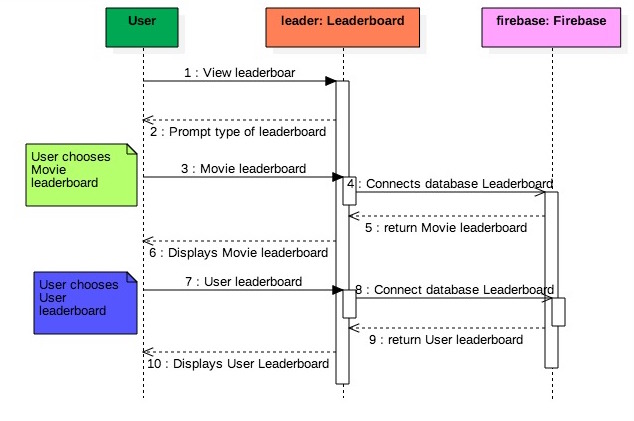
## Critique a Movie



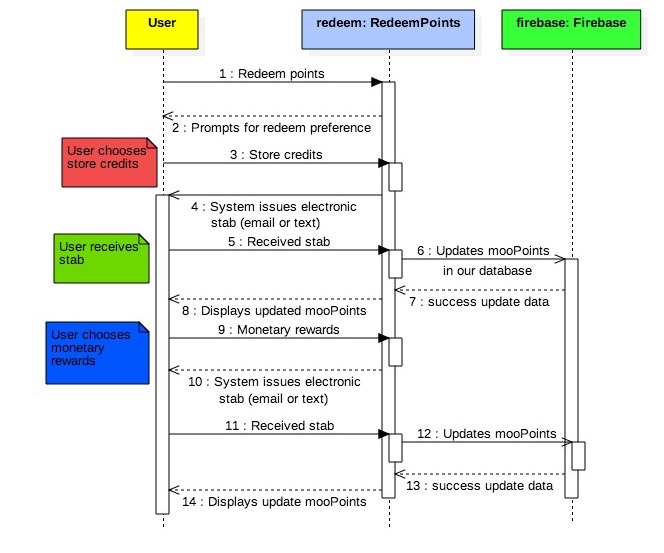
## Critique a Movie Trailer



## Leaderboard

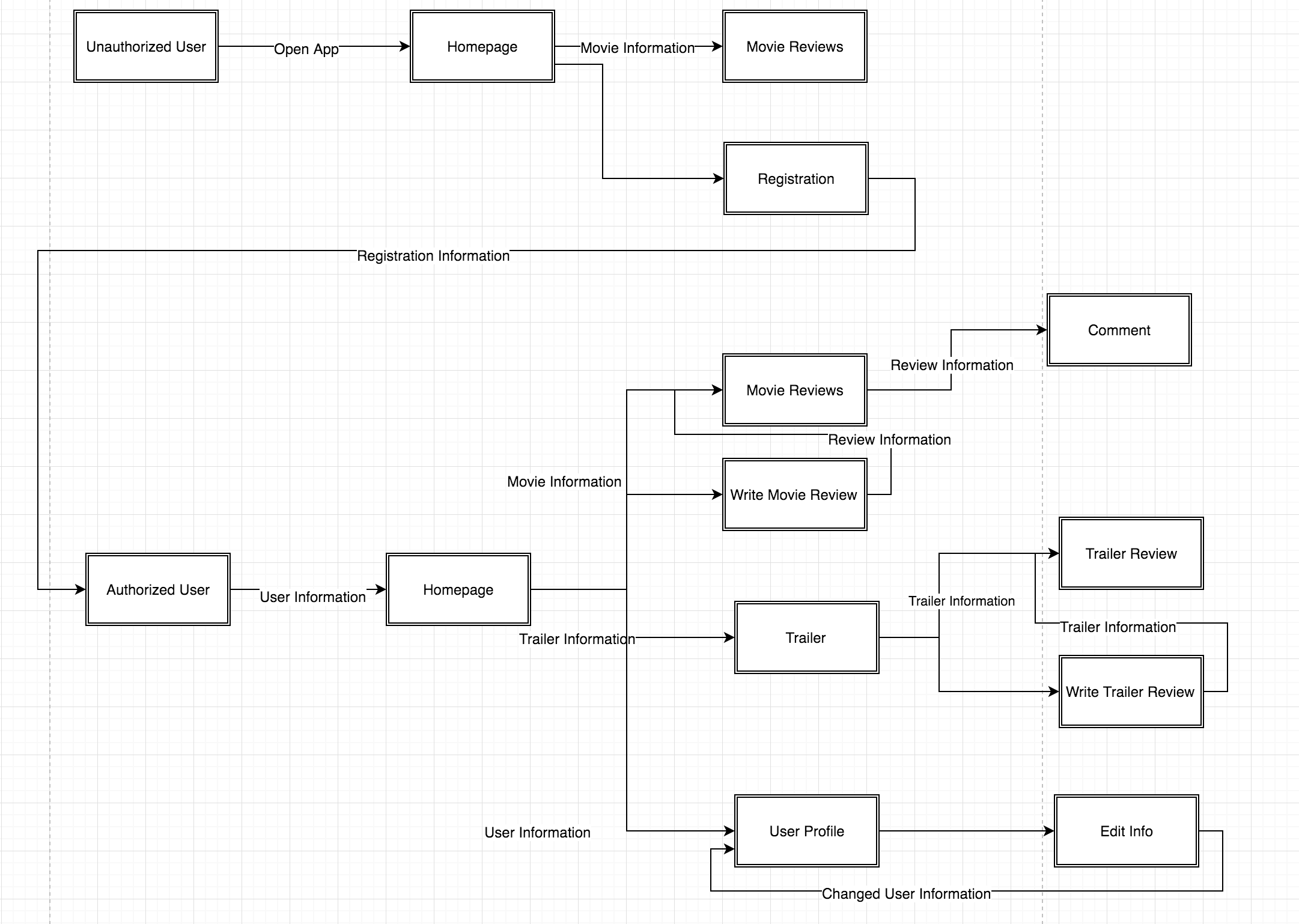


## Redeem Points



## 

# Data Flow Diagram

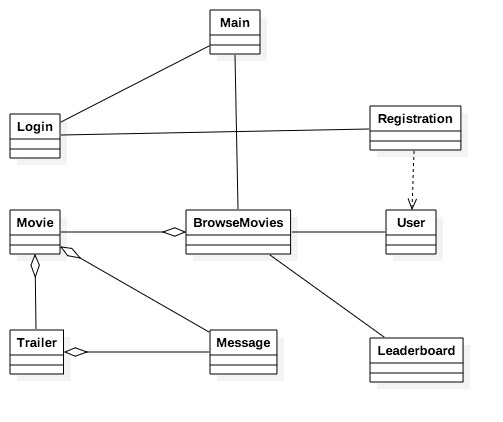
The diagram below describes the flow of data as unauthorized and authorized users. If a user opens the application for the very first time, the user is deemed an unauthorized user; he will have access to the movie information and reviews but will not have any privilege of liking, commenting, or writing reviews. Between each page in the application, there may be data required to be transferred in order to complete the user’s task. Movie information will consist of any information about the movie (name, length, rating, actors, etc) as well as the address of the reviews stored in the database to load these reviews. Trailer information is similar to movie information, except it will be for the trailer and will include the address to the trailer to be played. User information will consist of the user’s name, while the registration information will have the user’s name, date of birth, email, and password. The diagram illustrates how exactly data within the mobile application Moo flows into, within, and out of the application itself. It details how data flows among the application’s users, how the data is stored, and how data is processed internally. Each rectangle shows a data processing unit. In addition to the aforementioned and showcased sequence diagram, a data flow diagram is an extension to the detailed design of the mobile application Moo. 

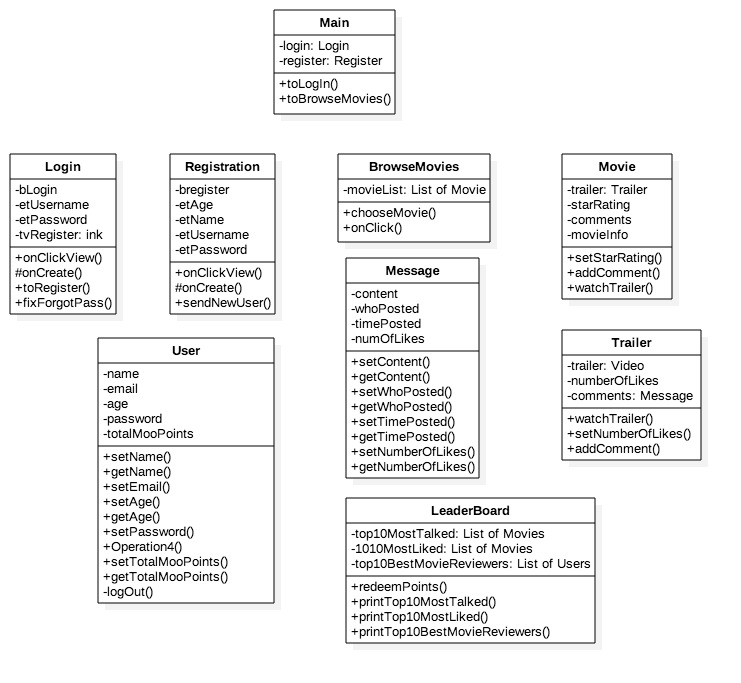
# 

# 

# Class Diagram

The below diagram helps comprehend and visualize the design of the mobile application Moo. It is the foundation of the application itself and will help the team develop each portion of the application. A sequence diagram will elaborate upon each of the classes below. The diagram depicts object types, attributes, and applicable operations as well as relationships between them. For example, a relationship occurs between registration and the user.





# 

# 

# Architectural View of Deployment

The architectural deployment view below showcases the mobile applicable Moo constraints as well as how maintenance will be handled upon finishing and releasing the application. Additionally, the diagram below denotes relationships between each individual architecture and what is required, for example, an internet connection to download the mobile application from the store.

# 

# 

# 

# Traceability Matrix

The matrix depicted below helps the team track each requirement with the applicable code to implement during each stage. The team can track each specific part of the code back to the applicable and corresponding elements of the design and requirement.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Requirements |  | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | 2.6 | 3.1 | 3.2 | 3.3 | 3.4 | 3.5 | 3.6 | 3.7 |
| Test Cases | Total | 9 | 9 | 7 | 1 | 2 | 1 | 1 | 2 | 4 | 2 | 2 | 2 | 8 |
| A password shall contain an uppercase, punctuation, and is at least 8 characters. | 1 |  |  |  |  |  |  |  |  |  |  |  | x |  |
| All users can read any movie review | 4 | x | x |  |  |  | x |  |  |  |  |  |  | x |
| Authorized users and administrators can create reviews | 4 | x | x | x |  |  |  |  |  |  |  |  |  | x |
| Authorized users and administrators can like reviews | 6 | x | x | x |  |  |  |  | x |  |  | x |  | x |
| Authorized users and administrators can comment on reviews | 7 | x | x | x |  |  |  |  | x | x |  | x |  | x |
| Authorized users and administrators can view user profiles | 7 | x | x | x |  |  |  |  |  | x | x |  | x | x |
| Authorized users and administrators can flag user reviews | 4 | x | x | x |  |  |  |  |  |  |  |  |  | x |
| Administrators can delete reviews | 2 |  | x |  |  |  |  |  |  |  |  |  |  | x |
| Administrators can ban users | 2 |  | x |  |  |  |  |  |  |  |  |  |  | x |
| Upon creating a review, a picture of a ticket stub is prompted | 3 | x |  |  |  | x |  |  |  | x |  |  |  |  |
| User will be remembered upon opening of the app if already logged in previously | 1 |  |  |  |  |  |  | x |  |  |  |  |  |  |
| Reviews shall consist no more of 1000 words | 2 |  | x | x |  |  |  |  |  |  |  |  |  |  |
| Authorized users can watch trailers | 1 | x |  |  |  |  |  |  |  |  |  |  |  |  |
| Authorized users and administrators can check the leaderboards | 1 |  |  | x |  |  |  |  |  |  |  |  |  |  |
| Authorized users can redeem their points | 1 |  |  |  |  |  |  |  |  | x |  |  |  |  |
| Authorized users and administrators can change their avatars | 2 | x |  |  |  |  |  |  |  |  | x |  |  |  |
| Users can run Moo on Android phone | 2 |  |  |  | x | x |  |  |  |  |  |  |  |  |