CS490 Design Documents
Team FireUp
Di Yao, Hanxun Huang, Kai Sun, Yijie Wu

Table of content

Purpose

Today, there are all kinds of social networks, but none of them are specifically designed for gastronomist. Therefore, we would like to create one. Our goal is to develop an application that continently for user to share their passion for food. We will build our applications on iOS and use server to support the application. User will be able to share their recipes and chat with other user.

General Priorities

1. Usability

Our main priority is usability. We want to implement this application easy to use for users. Using simple design style, every button on the page can give users a clear understanding what is the function of that button. Keep the entire UI as clean as possible.

2. Extensibility

This application should not just have limited to use as a platform for user to share their interest in culinary art. It can be extended to use as an application for users to meet and get to know other people who share same interest with them. For example, add function that allow user to find other user who want to have dinner or lunch together in local area.

3. Security

When sharing things online, privacy is very important. The location sharing function can also be turned off is users are not willing to share their locations. User can choose what kind of profile information to share with other user and this information are specific do not share with anyone.

4. Performance

Performance is another important property in our application. We will make sure the connections with the server are as fast as possible. We will user reliable server provider for iOS project and API.

Design Outline

Our system will use the Client-Server-Database architecture with a central server that communicates between client (i.e. UI) and database. It interacts with users by following steps: client sends user's request to server, server forwards requests to database and send the updated data that is retrieved from database back to client. We will implement this application (Client) and use Parse as server and database.

1.Client

Purpose: The client which is our application will implement with Swift and running on iOS devices. User may send request to the server, such as request to update feed, request to update personal information, request for instant message.

Interactions: The client send request to the server and receive data from the server. We will use Parse API in the implementation.

2.Server and database

Purpose: Server is a bridge between the application (client) and the database. It provides a function or service to one or many clients, which initiate requests for such services and forward the updates to database.

Interactions: Server receive request from client and data from database. Server can send data back to the client and request data from database. We will use Parse as our server and database. We will use Parse API in the implementation.

Figure1. UML diagram of high-level structure of our system

Design Issues

Issue 1: Parse vs custom server

Option1: Use Parse as server

Option2: Design and implement a server.

Decision: Option 1. Because designing and implement a server is time consuming and is not the focus of this project, instead we want to focus ourselves on creating the amazing app and bring good user experience, rather than spend lots of time on the complex server.

Issue 2: User vs Multiple types of User

Option1: Single User type

Option2: Multiple User type.

Decision: We will use option 2 since we will have different types of user. For example, we have users that are not professional chief that are just passionate about food and would like to share their experience or recipes with others, and we will also have users who are professional chief that using this app.

Issue 3: Users location share vs Users not sharing location

Option1: Users share location when make a post

Option2: Users not share location when make a post

Option3: Users can choose if they want to share location

Decision: Option 3. We believe the user's privacy are crucial to us, so it's not a good idea to share the user's location each time when they are posting something. But also, we want to make this app a fun app that you cannot only just share your recipe with other people, but also meet new people who are passionate about food like you, we decide to let our users make the call.

Issue 4: Post and comment for user, User login vs no user login

Option1: Allow un-login users can post or comment

Option2: Only login users can post or comment

Decision: We want to bring better user experience to our users. One thing we notice before is that if you allow un-login users to post stuff, usually, someone will post ads and irrelevant stuff, which is annoying for the users and lowers the grade for user experience.

Issue 5: How should users edit profile
Option1: Create profile page for each users
Option2: Users can edit profile when create account and not allow to change
Decision: Option 1. We allow users to edit their profiles anytime they want if their information has changed. It is a user friendly decision.

Issue 6: Using the application without internet
Option1: Notify there is a network error
Option2: Show cache contents which users already read before
Decision: Option 2. User can store their favorite recipe in locals. When they do not have internet connections, they can still get access to these recipes.

Issue 7: How to create account for professionals.
Option 1: Open registration in the application and verification process.
Option 2: Sending email to us to create account.
Decision: Option 1 and option 2. In this way, user can create an account and get access to the content of the applications. User can use the verification function within the application and also can send an email to support team of this project to get verification account.

Design Details

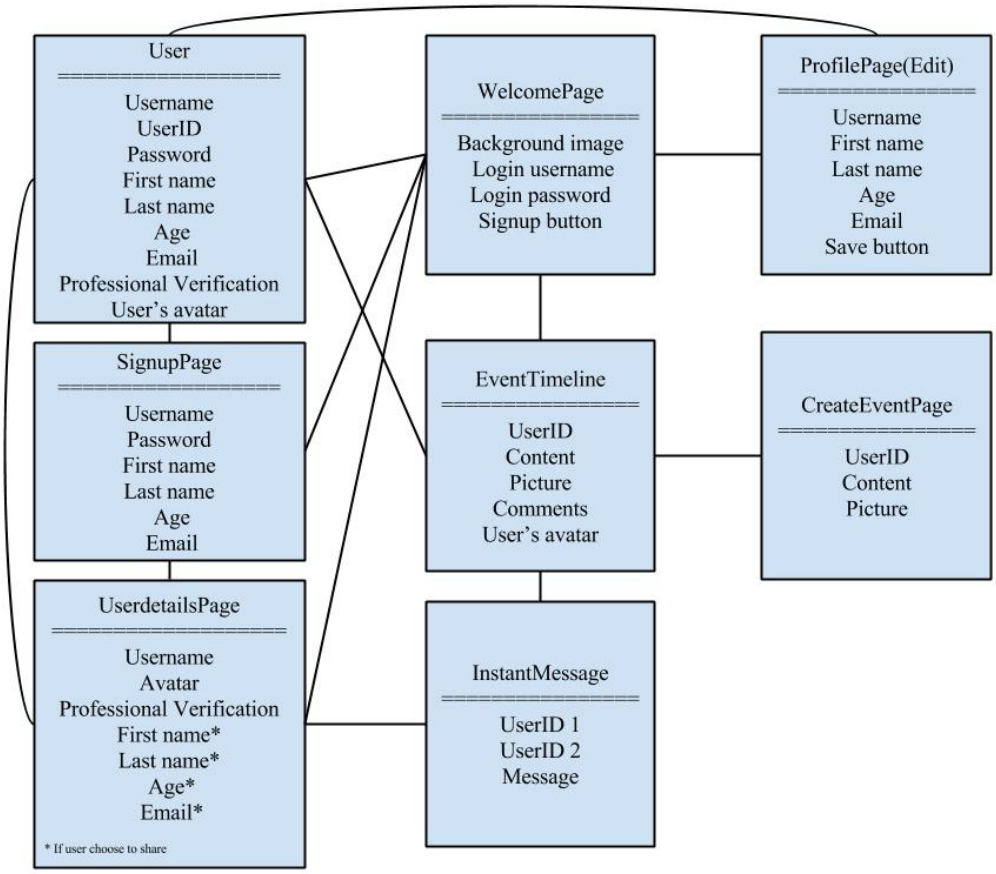

Figure2. Class diagram of the application structure.

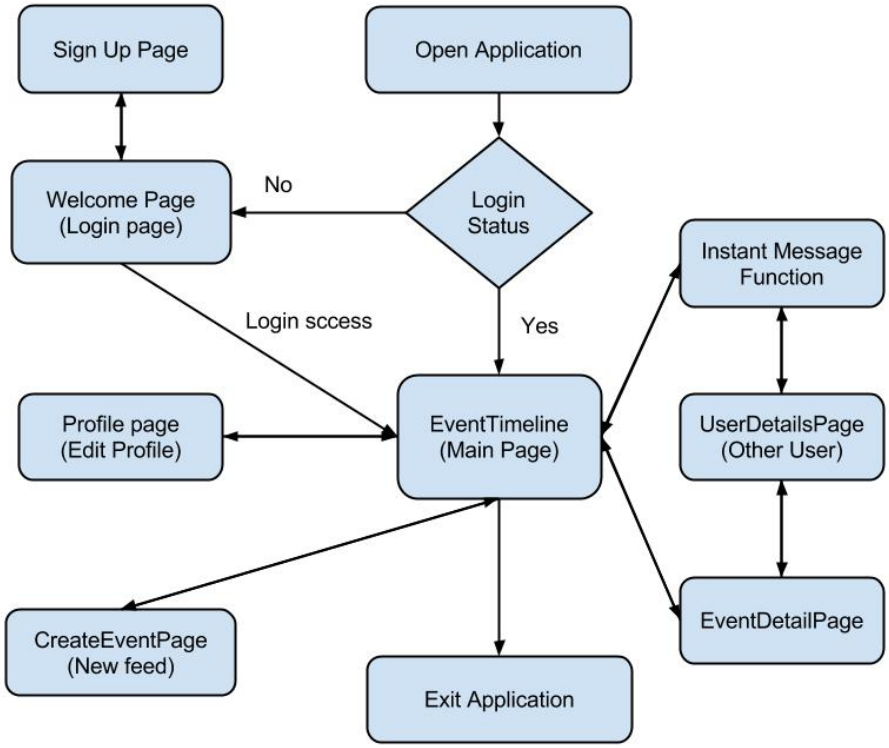

Figure3 Sequence diagram of the Application

Descriptions and interactions:

WelcomePage:

Attributes: Background image, login username login password and signup button.

Purpose: This welcome page will be the first look of our application. It will allow existed users login to their own account with correct username and password.

Interactions: Once users input username and password, system will check with database if there is correct pair with the input. If there is, users will successfully login and system will load this user's information and profile to the application. If users have not signed up yet, signup button will negative user to sign up page and users can sign up an account with us.

Users:

Attributes: username, first name, last name, age, email and password are set by users when create account or users also can edit later. Users without login can view detail post but can not leave comments or create instant message with other users. Users who want to post an event, talk to other users or leave comments need to sign up and login. Once users sign up, users can always login with this account.

Purpose: Every user is able to have their own account with their own information.

Interactions: Users' information will be stored in database and also interact within this class. Every time users change their profile, delete profile or load other's profile, this class will handle it.

EventTimeline:

Attributes: username, content, picture, comments and user's avatar.

Purposes: Display all the events to allow users view events by timeline in one page.

Interactions: EventTimeline page and every user have a one to many relationships. All users are allowed to view events within users' preferred range. Each event will be retrieved from our database within this page and stored in a data structure. As users go though each event, system will display from each single data structure to get data.

UserdetailsPage:

Attributes: username, avatar and all user details. User's information detail is only available if user allow other person to see it.

Purposes: One users find interested events. This page allows users to take look at user who post this event. And it will show all the information of the user who post this event. But if this event was posted by anonymously, this page will be disabled.

Interactions: When one user want to see other users' information, this page will load all information about the user if the event was not posted anonymously. It will first retrieve some key value from database to determine if this user can view detail information of the other user's information detail. If the answer's yes, it will load all information that are supposed to display and allow other users to view every details. And also users are able to create instant message within the page.

InstantMessage:

Attributes: two users' avatar and messages

Purposes: One users find themselves interested with each other, this will allow two users talk to each other in private.

Interactions: This function allows two users send messages to each other. Every message content will be stored in database. And every time a new message has been sent, we will use iOS notification to notify the other user.

ProfilePage:

Attributes: username, first name, last name, age, email and save button

Purposes: Allow user view his information and edit his own profile

Interactions: Once users click on "save" button, application will change users' profile and save it to database to allow other user view new profile.

CreateEventPage:

Attributes: When users create events, there are certain information that we need to complete the post. Users can choose if attach picture or if he wants to share his location. Also content is necessary for the post.

Purposes: Make sure users can post an event with enough details. And allow users to have privacy if users do not want to share location.

Interactions: Content of post, if users want to post anonymously and picture if users attach one will all be stored in database. Later when other users view other's posts, application will retrieve all this information and display.

SignupPage:

Attributes: Username, password, first name, last name and age are necessary when user want to create an account. Users can choose not enter email if users do want to.

Purposes: Accurate and enough information for users is necessary for displaying profile to other users and this will make users know each other well and have better interaction.

Interactions: All input information will be kept in database and be ready to retrieve whenever other users or users themselves view profile.