

Gamify

Software Design

CSCI-P465/565 (Software Engineering I)

Project Team

Meghesh Patil

Sriharsha Kaluva

Prashanth Bhat

Sagar Suresh Panchal

1. Introduction

This section introduces the design approach to the software system.

1.1 System Description

Gamify aims to create a social network for all those who love to play any sport or outdoor activities. It will allow people to co-ordinate so that they can play together if they are in the same location and can organize events. Users will be able to search people who have same interests of sports and can collaborate as per their schedule.

1.2 Design Evolution

This section is intended to document the rationale behind the selected design solution.

1.2.1 Design Issues

Application only needs a Web browser to be accessed.

1.2.2 Candidate Design Solutions

We have decided to build most of the components in the system using PHP as the backend. For chatting feature we will use Node.js. We are looking to deploy the application on the IU server using IU web hosting.

1.2.3 Design Solution Rationale

As PHP is a server-side web programming language that is widely used for web development, it has lot of documentation available over the net which would be helpful for us in case we run into coding issues.

IU web hosting is a reliable medium to host php scripts on the IU server. There are sources available in the kb.iu.edu to assist in hosting.

1.3 Design Approach

1.3.1 Methods

Until now we have designed the Login and User registration feature for which the approach used is that along with the normal registration and login process, we will encrypt the user's password and store it in the DB. Also, there will be a Duo authentication for all the users every time they login. Forgot password feature will consist of two options to reset the password i.e. either using Security Question or Duo authentication. OAuth is implemented for FB and Google using their own API's.

Once the user logs in, he/she will be able to post event posts, interact with other users using chat feature, select preferred sport/gender specific posts etc.

For testing, we would be using unit and regression testing. Unit tests are for stability and code predictability, while regression testing is for evaluating of how users interact with the code.

1.3.2 Standards

Passwords are encrypted using SHA512 algorithm, which uses a random salt to hash the password. After hashing, the hashed password and the corresponding salt both are stored in the DB to verify during the next login and authenticate the user. This is standard security protocol regarding passwords. We are adhering to the PHP standards and practices as mentioned in the official documentation. The core component in PHP are known as functions which is a logical collection of code which implements a specific feature of our website.

Apart from these, we are using Bootstrap 3's standards and practices to design our HTML pages. Also, CSS 3 will be used to provide better look and feel of the GUI

1.3.3 Tools

- Editor – Phpstorm, Sublime, Atom, Notepad++
- Webserver- Xampp, Wampp
- Database - MySQL
- Hosting- IU web hosting

2. System Architecture

2.1 System Design

At a higher level, our system will have classes/functions for each component implementing a specific feature. At a lower level, each functions will have its own Logic, URL Routing, and Unit Tests.

We have used a basic template for our HTML pages having the same header and footer. This template will be inherited for all the HTML pages that we develop.

2.2 External Interfaces

We are using OAuth using a third party Facebook API as well as Google API with the help of [Facebook for Developers](#) and [Google API's](#) respectively. Users can login and register using their Facebook / Google credentials.

3. Component Design

Component Name (1)

Login and registration

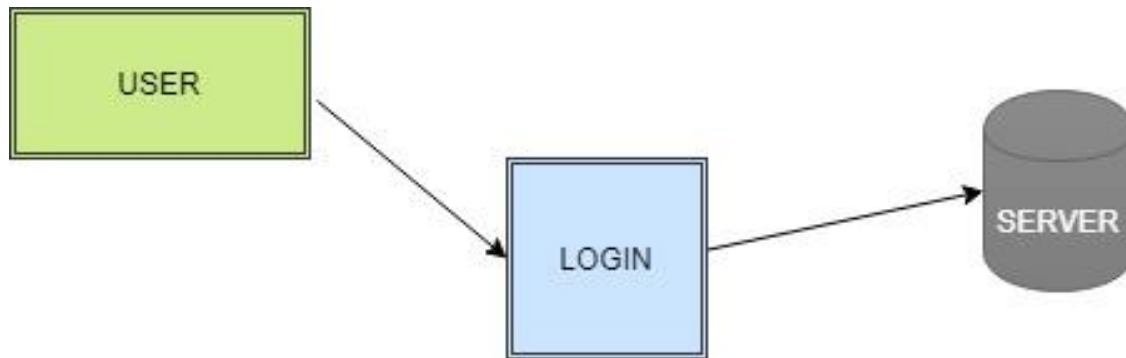
Component Description

The Login form allows the users to login with the credentials. When providing the correct credentials, the user will be receiving Duo authentication mail containing an One Time Password (OTP). Then the user must authenticate for the second time using this OTP and once they are verified, they will be logged in to the website. Once the user is logged into the website, the website provides an option to logout of the web page. We have also implemented OAuth using a third party API such as Facebook / Google to enable users to directly register with their Facebook / Google credentials. The registration form will allow users who don't want to use OAuth to register by filling in their details.

Responsible Development Team Member

All the team members had divided this major component into smaller chunks and then integrated it together.

Component Diagram (1)



Component Objects

User Component: A User who will be connect to the URL of the Website and then will either register themselves if they will be new user or else they will log in to the site if they are existing users.

Login Component: This component consists of multiple objects which are given below

Login – Authenticate the User based on provided credentials

Registration – Form to register new user along with the security questions.

Duo Authentication – Send an OTP to the user in email and authenticate once they have provided correct OTP.

Forgot Password – Forgot password has 2 options, Security question and OTP. Security Question will have 2 questions, if user answers them correctly they will be able to reset the password. Or an OTP will be sent to their registered email, if they authenticate themselves using OTP they will be able to reset their password.

Database Component: This component will store all the User's data and credentials and will be used to authenticate the users based on the Database calls made to the server.

Component Interfaces (internal and external)

The component will make calls to the Database to validate the credentials of users trying to access the system. The component will check with the third-party Facebook / Google API to validate the credentials.

Component Error Handling

If a user is trying to login with invalid credentials such as Invalid username or Invalid password, they are handled in the component.

Component Name (2)

Create posts and display posts

Component Description

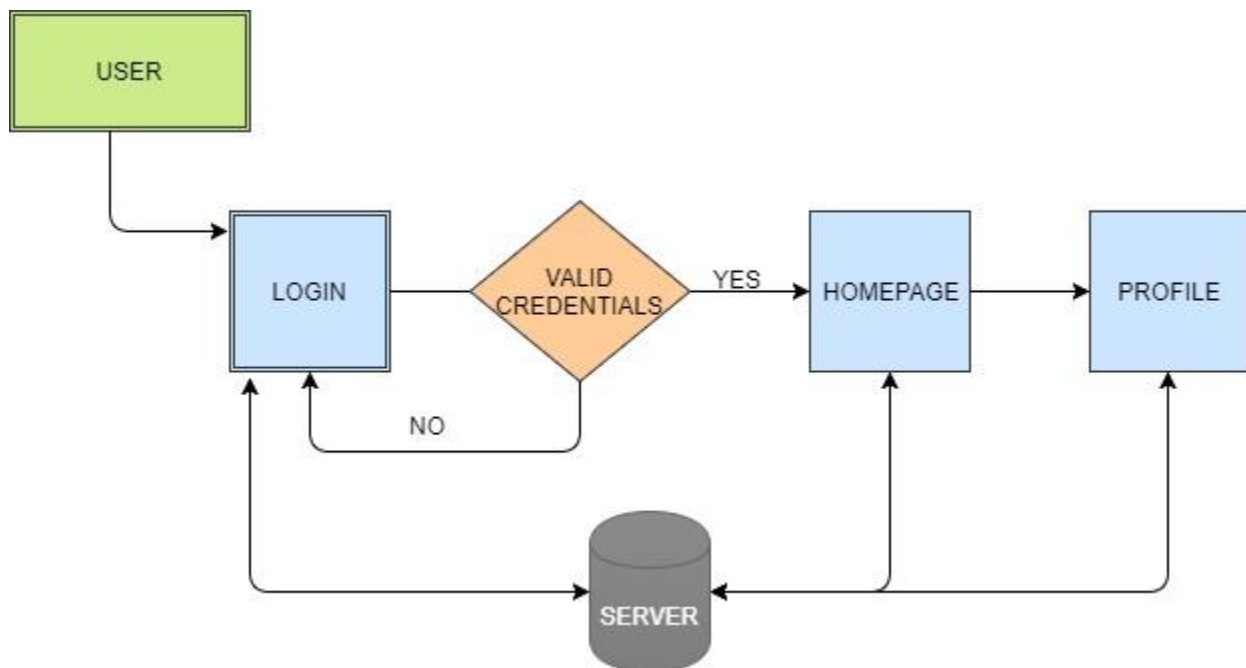
The create post allows the user to create a post that they want to display. The user can select the Location, Event name, timing and the gender for that event. On clicking on the post button, the user should be able to see his posts and all other user with the same preferences should be able to see the post as well. On viewing the post, the user can like the post by clicking on the like button or comment on the post. The comment on that post will also be displayed.

The user can also input their preferences and according to their preferences the data will be displayed. The user can navigate back and forth User profile page.

Responsible Development Team Member

All the team members had divided this major component into smaller chunks and then integrated it together.

Component diagram (2)



Component Objects

Flow until Login is described in Component diagram (1)

Homepage component – The user will be redirected to the home page where they will be able to see the posts.

- Create Post– Here the user is able to create post as per the sport, location gender and time of the event.
- Display Post– The post by user will be display on their wall.
- Comment– The user should be able to comment on their post or somebody else's posts.

- Like– The user should be able to like the post that he sees.

Profile page component-

- Profile picture- Display picture the user has uploaded.
- Personal details- First name, Last name, Email ID, phone number, etc.
- User posts- The posts that the user has posted.

Database Component: This component will store all the User's data and credentials and will be used to authenticate the users based on the Database calls made to the server to redirect the user to the homepage and profile page if required.

Component Interfaces (internal and external)

The component will make calls to the Database to get newsfeed, user profile pages.

Component Name (3)

Messaging

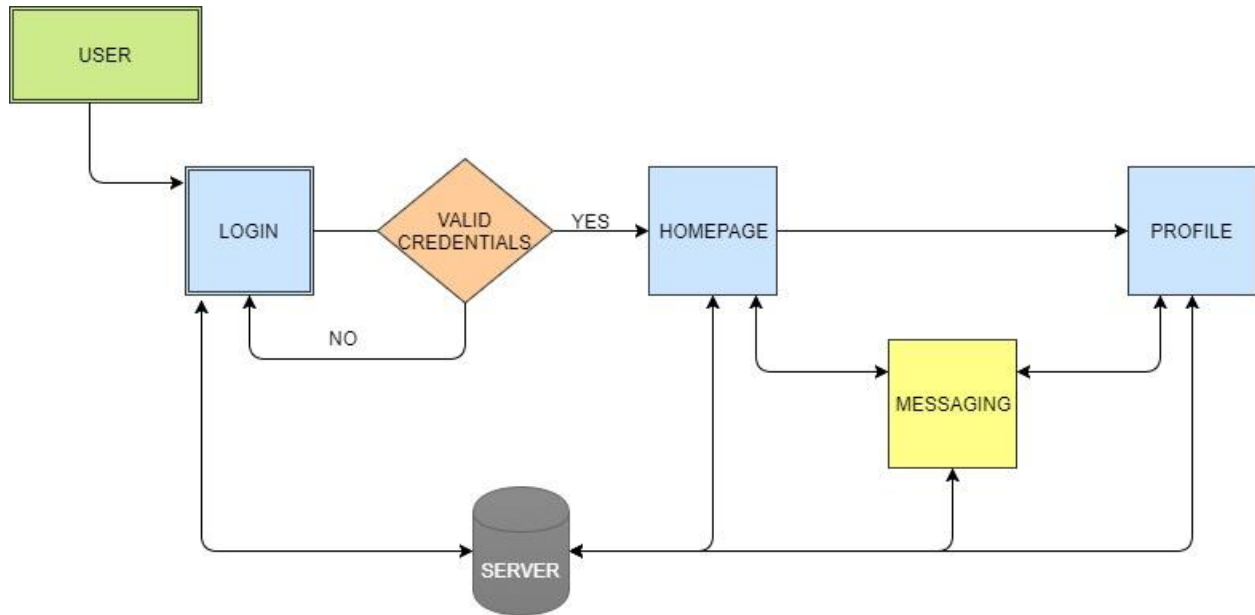
Component Description

Once the user is in homepage or profile page, he can navigate to the messaging page where he has to enter the email ID or user name to send a 'message request'. If the entered username is invalid or not found, no search results will be shown. If exists, the user will be able to send the request. Once the other user accepts, they'll be able to exchange messages.

Responsible Development Team Member

All the team members had divided this major component into smaller chunks and then integrated it together.

Component Diagram (3)



Component Objects

Flow until Homepage is same as in Component Diagram (2).

Messaging component- A user can send/ receive messages from public who're registered on the website. However, a user has to send a message request to other user and when the second user accepts the request, only then the first user can send/receive messages from the second user.

Database Component: This component will store all the message exchanges and conversations based on the Database calls made to the server.

Component Interfaces (internal and external)

The component will make calls to the Database to send a message request to other user. The stream of messages are exchanged with each call to and from the database.

Component Error Handling

If a user is trying to enter a username which is invalid or not found (doesn't exist), no search results will be shown.

Revision History

Revision	Date	Change Description
Initial Version (V.1)	10/01/2017	Initial Draft
Version (V.2)	10/15/2017	Sprint 2 Update
Version (V.3)	10/29/2017	Sprint 3 Update