Koobecaf

Software Design

CSCI-P465/565 (Software Engineering I)

Project Team

Mohit Murali Alumullithody

Tony Dominick Dattolo

Rushikesh Pharate

Rahul Gopichand Shamdasani

1. Introduction

Agile Methodology will be followed to ensure that all the requirement of the customer are met on time and the application works as expected. The team will follow component-based design approach for code-reusability purposes.

1.1 System Description

The team plans to create a social media app which will allow users to create profiles, post images, videos etc. and chat with friends. The team believes that this application will be of significant help to people amid the Covid Pandemic. The team will be focusing on ensuring that the website is clean and intuitive has user-friendly user interface. The team will be delivering the features in agile model, thus incrementally over the period of three months. The technology stack involved will be Django(backend) and React JS(frontend).

1.2 Design Evolution

To ensure that all customer get hassle free experience the team will be focusing towards ensuring that UI component is as simple as possible. All the user must be able to easily login, socialise, post, chat with their friends through the application. To ensure this the team decided to break the core features across various sprints.

1.2.1 Design Issues

Since the project is to be hosted on the internet, A reliable internet connection is expected to use it. Apart from that, The team is expected to implement OAuth 2.0

for Authorisation , which enables users to login through their registered google accounts.

1.2.2 Candidate Design Solutions

Stable Internet Connection : The website will ensure that the users are connected with their friends real time.

OAuth 2.0 : The integration with OAuth 2.0 lets users to create account using their gmail credentials.

Search : The users can search for their interest, friends through the use of search functionality provided in the website.

Dark Mode: The users can switch between two themes for the entire application, One will be Light mode(default mode) another would be dark mode.

1.2.3 Design Solution Rationale

The intention behind using component based approach is that it ensures that the code maintenance cost is low. With proper documentation, code could be reused across various modules as well which would ensure that the team faces no hurdles in development.

1.3 Design Approach

1.3.1 Methods

The team will be following Agile standards during software development and will strictly adhere to agile standards like daily standup, sprint retrospective etc. Each sprint comprising of 2 weeks will be focussed towards developing a working prototype which would then be demonstrated to the client. Component based design pattern will be employed for code maintenance and reusability purposes.

1.3.2 Standards

The team is ensuring that the website developed has clean UI, intuitive and user friendly. The Team is closely following all standard design standards across the market to ensure a seamless experience is provided to the customers.

1.3.3 Tools

FrontEnd : React JS, Next JS
Backend : Python (Django)
Databases : PostgreSQL
Authorisation : OAuth 2.0

2. System Architecture

2.1 System Design

Since the project is a website, Users interact with it through a browser. The front-end is handles by React, which is popular Javascript framework and is integrated tightly with Django framework of python. All the API endpoints are handled by Django framework. During registration process, User can enter details manually which will be saved by the backend in the server or via OAuth. In OAuth, as User clicks on authenticate, corresponding email and password are shared to Google API via OAuth which validates the credentials and returns the basic information of the user along with access tokens to backend ,which then saves it and redirects the user to homepage. In a similar way login functionality is also implemented.

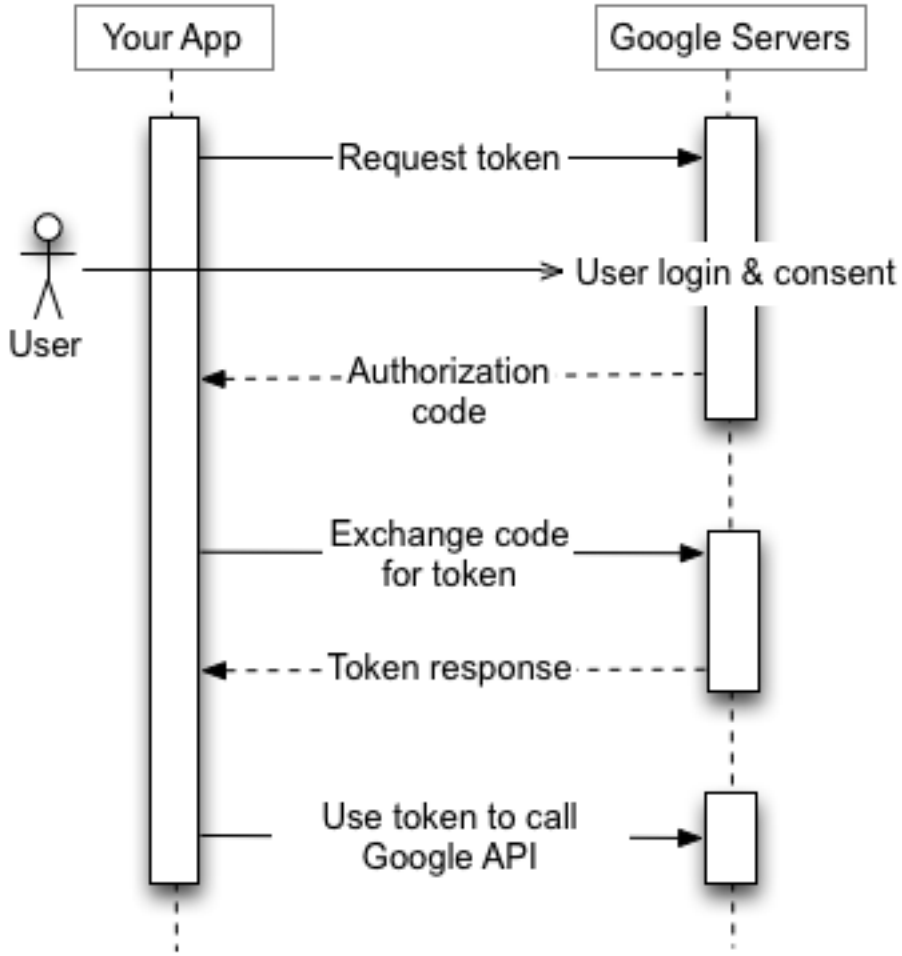


Figure 1 : OAuth Flow, Reference : https://developers.google.com/identity/protocols/oauth2

2.2 External Interfaces

Django Framework will be used for implementing backend base code for API calls.
OAuth 2.0 API will be used for Authentication purposes, specific to google accounts.

3. Component Design

- **Component Name**
  Authentication Component
  Posts Component
  Chat Component
  Group Pages Component
  Search Component
  Marketplace Component
  Profile Component
  Theme Component

- **Component Description**
  Authentication Component will be used for registering and logging into the website. It has basic activities involved in logging into any website. It also uses SSO functionality where user can signup/login using their google credentials.

  Posts Component aims where once user has logged in can share anything(text or media) through a post on the website.

  Chat Component will be used by users to interact with each other through normal text conversations.

  Group Pages Component will be used by users to join a popular group where in they can view posts and share posts which will be shared by all the members of that group.

  Search Component will be used by users to search on specific user or group page of their interest.

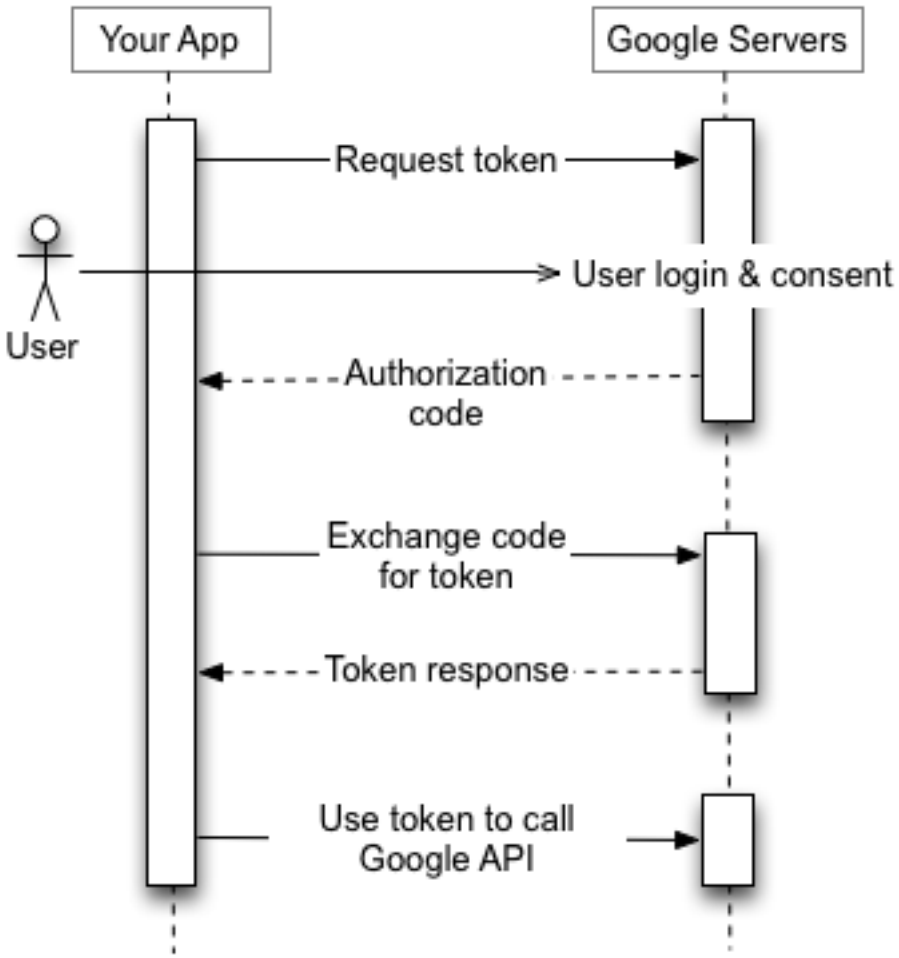
  Marketplace Component will be used by users to posts item which they wish to sell on the website. It will contain basic information of the item on sale.

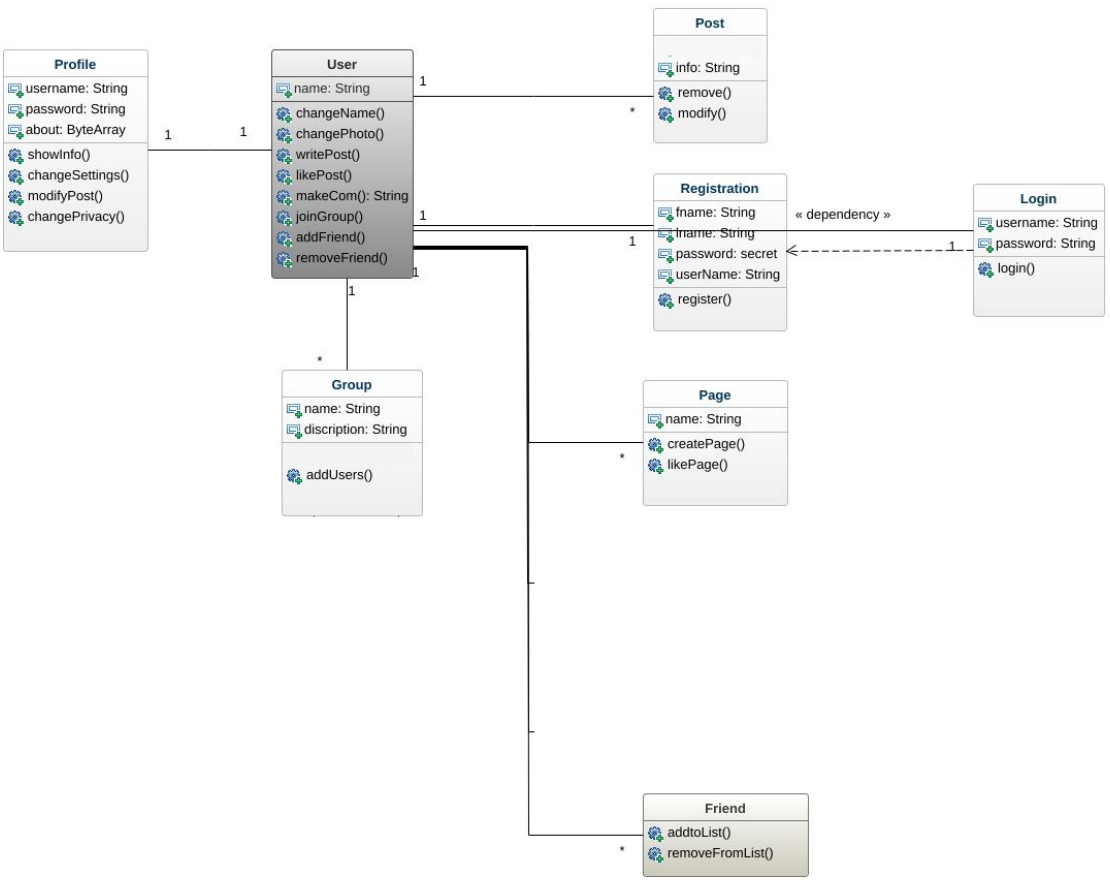
  Profile Component will be used by users to view the basic information regarding their profile

  Theme Component will be used by user to toggle between Light and Dark mode respectively.

- **Responsible Development Team Lead Member**

  Authentication : Mohit
  Posts Component : Tony
  Chat : Rahul
  Group : Rushikesh
  Search : Tony
  Marketplace : Rushikesh
  Profile : Mohit
  Dark Mode : Mohit


Your App
Google Servers
Request token
User
User login & consent
Authorization code
Exchange code for token
Token response
Use token to call Google API


Profile
username: String
password: String
about: ByteArray
showInfo()
changeSettings()
modifyPost()
changePrivacy()
User
name: String
changeName()
changePhoto()
writePost()
likePost()
makeCom(): String
joinGroup()
addFriend()
removeFriend()
Post
info: String
remove()
modify()
Registration
fname: String
lname: String
password: secret
userName: String
register()
« dependency »
Login
username: String
password: String
login()
Group
name: String
discription: String
addUsers()
Page
name: String
createPage()
likePage()
Friend
addtoList()
removeFromList()

- **Component Diagram**
-
- **Component User Interface**
  The UI for all the components are HTML Pages. In authentication component, page is similar to a form.

- **Component Objects**

  - Authentication component will be called when user tries to login to the website either with credentials or with SSO functionality(i.e. Google accounts). It's also used during the user registration process(Email verification). It will also incorporate reset password functionality through email verification.
  - When user tries to login through SSO, a call is made to Google API via the OAuth server, which validates the users and return and redirection url(Homepage in our case if success) along with access tokens and refresh token which are stored in the backend of the system by Django framework.
  - Profile component is responsible for loading the basic information page about the user after the user is logged into the system.
  - Group component is responsible for loading the basic information page about the specific group which users has searched, after the user is logged into the system.
  - Search component is called when logged in users search any specific person or page from search bar.
  - Chat component is used for interaction between users through text conversation.
  - Marketplace component is used for displaying relevant items on sale. It is similar to a post with more details of Amount, date posted etc.
  - Posts Component is for ensuring that users can post text or media content onto the system.
  - Theme Component will be used by user to toggle between Light and Dark mode respectively.

- **Component Interfaces (internal and external)**
  The UI communicates with the backend API which instead communicates with OAuth 2.0 server for authentication purposes.

- **Component Error Handling**
  During Signup process, if the user has not entered password which meets the general password format standards, an error will be reflected. During login, if the user enters incorrect email address or password, an error reflecting the same will be shown to the user.

Revision History

Revision	Date	Change Description
1.0	2nd October	Initial Version
2.0	17th October	Modified to incorporate changes from Sprint 2
3.0	25th October	Modified to incorporate changes from Sprint 3
4.0	14th November	Modified to incorporate changes from Sprint 4

Page Author : Mohit Murali Alumullithody
Last Modified : 11/14/2021