System Design

CSCC01 Summer 2022

Final Project

Team: Amorr Yes

Lingfei Cai, Zhenyuan Xiang, Changhao Wang,

Zhongyu Liu, Pei Zhang

Contents

CRC Cards

NavBar	
Parent Class: App Subclass: None	
Responsibilities: Enables users to navigate to one of the 3 primary pages of the website, or to log out. Navigation button availability depends on the login state of the user. It always stays at the top of the webpage and does not interfere with the main body.	Collaborators:

Body	
Parent Class: App Subclass: None	
Responsibilities: Serves as a container for the main body of each webpage. Allows redirection to different pages without actually reloading the entire page.	Collaborators: ● SignUp ● Login ● ClientProfile ● ClientIDUpload

App	
Parent Class: None Subclass: None	
Responsibilities: Runs the frontend application. Simply combines and displays NavBar and Body.	Collaborators: ● NavBar ● Body

SignUp	
Parent Class: Body Subclass: None	
Responsibilities: Displays the signup page. Allows users to enter signup information and submit it to the server for account creation.	Collaborators:

Login	
Parent Class: Body Subclass: None	
Responsibilities: Displays the login page. Allows users to enter login information and submit it to the server for verification.	Collaborators:

ClientProfile	
Parent Class: Body Subclass: None	
Responsibilities: Displays the client profile page. Allows users to view and edit their profile.	Collaborators:

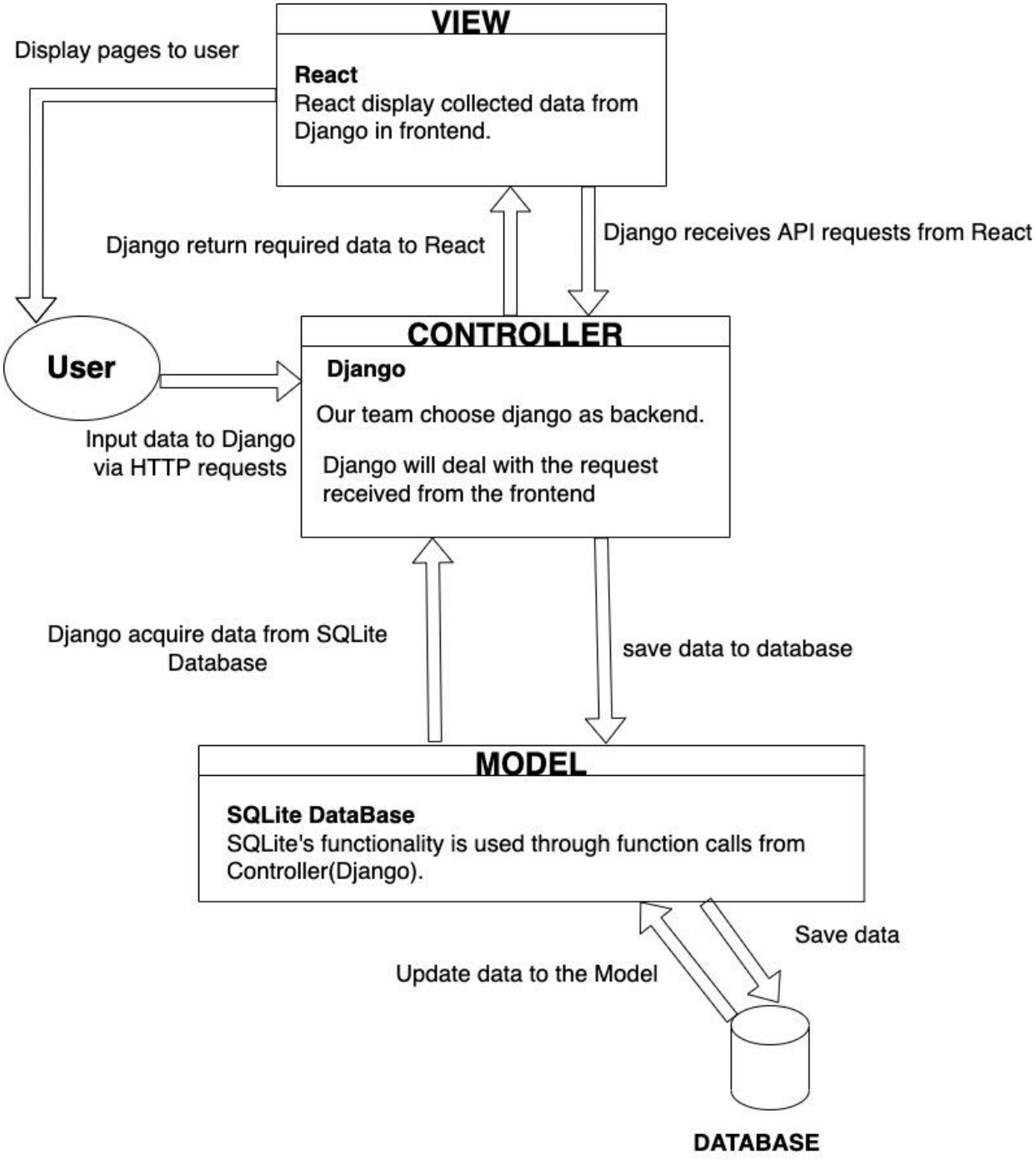
ClientIDUpload	
Parent Class: Body Subclass: None	
Responsibilities: Allows users to upload and reupload their photo ID to the server. Display the ID if already uploaded.	Collaborators:

Database table schemas

User	
id	integer
first_name	varchar(60)
last_name	varchar(60)
email	varchar(100)
password	varchar(100)
about	text
photoid	varchar(200)
phone	varchar(100)

System Architecture

We use MVC architecture for our project. Architecture Description and System Interaction Description are included in the diagram.



System Decomposition

Frontend:

The user can send api requests through the presentation tier. The frontend is responsible for gathering and displaying information. React will display the signup/login forms and send the information(each max length is 32 letters) as json to the frontend. Before redirecting to other pages, the frontend will send a request to the backend to check whether the user is logging in. If not, the frontend will redirect to the login page.

Backend

After Django receives the requests, it deals with the request and then query the SQL database. The application layer will then send http requests to the frontend so that it can display information to the user. If the backend receives an empty username, it will return status failure and false. Other errors like network failure will be thrown to exception.

SQL database

SQLdatabase will store the data it received and send back the data to Django. We choose to add salt into a MD5 which can help protect our information.