GeoCompensate

CSE-5324-001 Software Engineering: Analysis, Design and Testing

Iteration - 3
Date: 24th April, 2024

Group 8: CodeCrafters

Team Members:
Param Shah
Hetvi Joshi
Jainik Gadara*
Nandish Patel
Sakshi Naik

Content

Project Description

GeoCompensate addresses the challenges businesses face in managing their mobile workforce efficiently by providing an innovative solution that seamlessly integrates cutting-edge GPS technology with automated payroll processes. The app aims to solve the problem of manual errors and inefficiencies in traditional payroll management. By automating payroll calculations based on real-time GPS data, GeoCompensate ensures unparalleled accuracy in tracking work hours, breaks, and overtime. The introduction of GeoCompensate responds directly to the need for fair and precise remuneration in varying geographical zones, allowing businesses to dynamically adjust employee compensation based on the specific locations where work is conducted. This app streamlines payroll operations and enhances transparency, efficiency, and accuracy, ultimately revolutionizing how businesses manage their mobile workforce.

Functional Requirements

Function 01: Register Employee

- **Role: HR -** HR has access to create the employee's account wherein the employee will receive their login credentials on their email ID.

Function 02: User Login

- **Role**: **HR** - HR acquires their credentials from the company's technical staff/ IT support.
- **Role: Employee -** The employee can access the system by logging in using the credentials they receive from HR on their email.

Function 03: Edit Profile

- **Role: HR -** HR has the ownership to modify their profile.
- **Role: Employee** - Employee has the ownership to modify their profile.

Function 04: Password Management

- **Role: HR -** HR can change the password of their profile if needed.
- **Role: Employee** - Employee, similarly has the right, at their discretion, to change the password that was initially issued.

Function 05: Clock In/Clock Out

- **Role: Employee -** Employee can clock in when within the designated geolocation zone, and clock out at the end of their shift to record work hours. The system continuously sends an employee's GPS location every 5 minutes to monitor and track their working hours accurately. If an employee leaves the premises without clocking out it will automatically clock out.

Function 06: View Timesheet

- **Role: Employee -** Employee has access to a report displaying their working hours for the last 30 days along with the pay for that period.

Function 07: View Employee’s Timesheet

- **Role: HR -** HR can view comprehensive reports of employee activities, including clock-in and clock-out times, for the preceding 30 days.

Function 08: Search Employee

- **Role: HR -** HR can search for specific employees based on their first name or last name. Search results based on the name will be shown in the table. If no records are available then an appropriate message will be shown.

Function 09: Update Employee

- **Role: HR -** HR has the authority to edit an employee's profile within the system enabling effective administration of the staff.

Function 10: Remove Employee

- **Role: HR -** Only HR has the authority to relieve the employee when an employee is fired or leaves the job.

Function 11: Logout

- **Role: HR -** After clicking on the logout button HR can log out from the app.
- **Role: Employee -** After clicking on the logout button employees can successfully terminate their clock-in time and will be able to log out from the app.

Resources

GPS, Wireless or cellular network, Database, APIs, and Permission management

Crafter Introduction

1.Param Shah: I bring extensive experience as a Full Stack Developer from my tenure at Fractal Analytics. I played a pivotal role in developing various modules for the company's website. My expertise encompasses both front-end and back-end technologies, utilizing Vue.js for the front end and Node.js with Express.js for the back end. I adeptly managed databases using MySQL and incorporated the Quasar framework to enhance website performance. My technical skills include proficiency in Java, JavaScript, Vue.js, and Node.js, along with database expertise in MySQL and MongoDB.

2. Hetvi Joshi: As a Software Engineer at Aubergine Solutions Pvt. Ltd. (December 2020 - July 2023), I successfully led a WordPress to Strapi CMS migration with React, improving flexibility and UI. I orchestrated key client modules in React and Node, and implemented AWS CRON job for resume filtering, boosting recruitment efficiency by 65%. Managing the company website project, I leveraged my Master's in Computer Science (the University of Texas at Arlington) for "Dutch N Settle," showcasing expertise in React, NodeJs, MongoDB Atlas, Google OAuth, PubSub, App Engine, Google Secret Manager, and Cloud Functions, achieving a 30% security improvement through Google OAuth integration.

3. Jainik Gadara: During my tenure at HHAeXchange (Dec 2020 - June 2023), I optimized database procedures and modernized legacy code. Achievements include integrating OAuth identity, developing an ETL CRON job using Hangfire, and revamping a Silverlight app with Angular and .NET 6. Holding a Master's in Computer Science, my "DutchNSettle" project showcased expertise in NextJS, Redux, NodeJS, MongoDB Atlas, and Google OAuth, improving security, data management, and notification efficiency. My skills span various languages, databases, and cloud services, highlighting versatile software development and engineering proficiency.

4. Nandish Patel: Currently pursuing a major in computer science at UT Arlington, I completed my undergraduate degree at Ahmedabad University, India, in fall 2023. I underwent a frontend development trainee course, to develop my skills in HTML, CSS, JavaScript, jQuery, and React. Notable achievements from my undergraduate program include the creation of Digiwheels, an iOS-based mobile application designed in Figma. This app facilitates user uploads and offers a multi-functional platform allowing access to transport-related information, appointment reservations, and convenient viewing of essential transportation documents like driver's licenses and vehicle registrations.

5. Sakshi Naik: Coming equipped with a comprehensive experience of Full Stack development, working as a React intern at Symphony Technologies Pvt Ltd, I engineered the integration of an AI-powered chatbot with a WordPress editor. Currently pursuing a major in computer science at UT Arlington, I completed my undergraduate degree in CSE from MIT WPU, Pune, India, in May 2023. I along with my teammates designed a cutting-edge PHP, HTML, and MySQL internship

database management system, streamlining data organization and accessibility; optimized query response time by 50% and facilitated data-driven decision-making processes.

Requirements

Req ID	Requirement Statement	Line reference
R1	The system shall provide functionality of employee registration to HR.	14 to 15
R2	The system shall provide login functionality to the users.	17 to 19
R3	The system shall provide an edit profile option to the users.	21 to 22
R4	The system shall provide a change password option to the users.	24 to 26
R5	The system shall provide an option to clock in, if within the work geolocation.	28 to 32
R6	The system shall provide an option to clock out, if within the work geolocation.	28 to 32
R7	The system shall provide an option to employees to view their timesheets.	34 to 35
R8	The system shall enable HR to view the timesheet of employees.	37 to 38
R9	The system shall enable HR to search for employees.	40 to 42
R10	The system shall enable HR to update an employee.	44 to 45
R11	The system shall enable HR to remove an employee.	47 to 48
R12	The system shall allow the users to logout.	50 to 52

Use Case List

Use Case #	Use Case Name
UC1	Register Employee
UC2	Login
UC3	Edit Profile
UC4	Change Password
UC5	Clock In
UC6	Clock Out
UC7	View Timesheet
UC8	View Employee’s Timesheet
UC9	Search Employee
UC10	Update Employee
UC11	Remove Employee
UC12	Logout

High-Level Use Cases

*TUCBW - The Use Case Begins With
*TUCEW - The Use Case Ends With

- **UC 1: Register employee**
    - TUCBW the HR clicks the “Register Employee” button.
    - TUCEW the HR sees the success message.

- **UC 2: Login**
    - TUCBW the user clicks the “Log In” button.
    - TUCEW the user gets successfully logged into the system.

- **UC 3: Edit Profile**
    - TUCBW the user clicks the “Edit profile” button.
    - TUCEW the user sees the “Profile has been updated successfully” message.

- **UC 4: Change Password**
    - TUCBW the user selects the “Change password” button.
    - TUCEW the user sees the “Password has been changed successfully” message.

- **UC 5: Clock In**
    - TUCBW the employee clicks the “Clock In” button.
    - TUCEW the employee sees the “Clocked In” message.

- **UC 6: Clock out**
    - TUCBW the employee clicks the “Clock Out” button.
    - TUCEW the employee sees the “Clocked out” message.

- **UC 7: View Timesheet**
    - TUCBW the employee clicks the “Timesheet” button.
    - TUCEW the employee sees monthly working hour details.

- **UC 8: View Employee’s timesheet**
    - TUCBW the HR clicks the “Timesheet”.

- TUCEW the HR sees the employee's clock-in/clock-out details day wise for that month.

- **UC 9: Search employee**
  - TUCBW the HR clicks on the search button after entering the employee first or last name.
  - TUCEW the HR sees the employee(s) records in a table format.

- **UC 10: Update employee**
  - TUCBW the HR updates the required fields and clicks on the “Update Employee” button.
  - TUCEW HR sees the “Employee profile updated successfully” message.

- **UC 11: Remove employee**
  - TUCBW the HR clicks on the “Relieve Employee” button.
  - TUCEW HR sees the “Relieved Employee Successfully” message.

- **UC 12: Logout**
  - TUCBW the user clicks the “Logout” button.
  - TUCEW the user clicks the “Yes” button and is logged out of the system.

Use Case Diagram

Use Case Traceability Matrix

	Priority weight	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
R1	1	X											
R2	1		X										
R3	4			X									
R4	5				X								
R5	2					X							
R6	2						X						
R7	3							X					
R8	3								X				
R9	2									X			
R10	4										X		
R11	4											X	
R12	1												X
	Score	1	1	4	5	2	2	3	3	2	4	4	1

NOTE- Priority 1 is the highest priority, work this first.

Increment Matrix

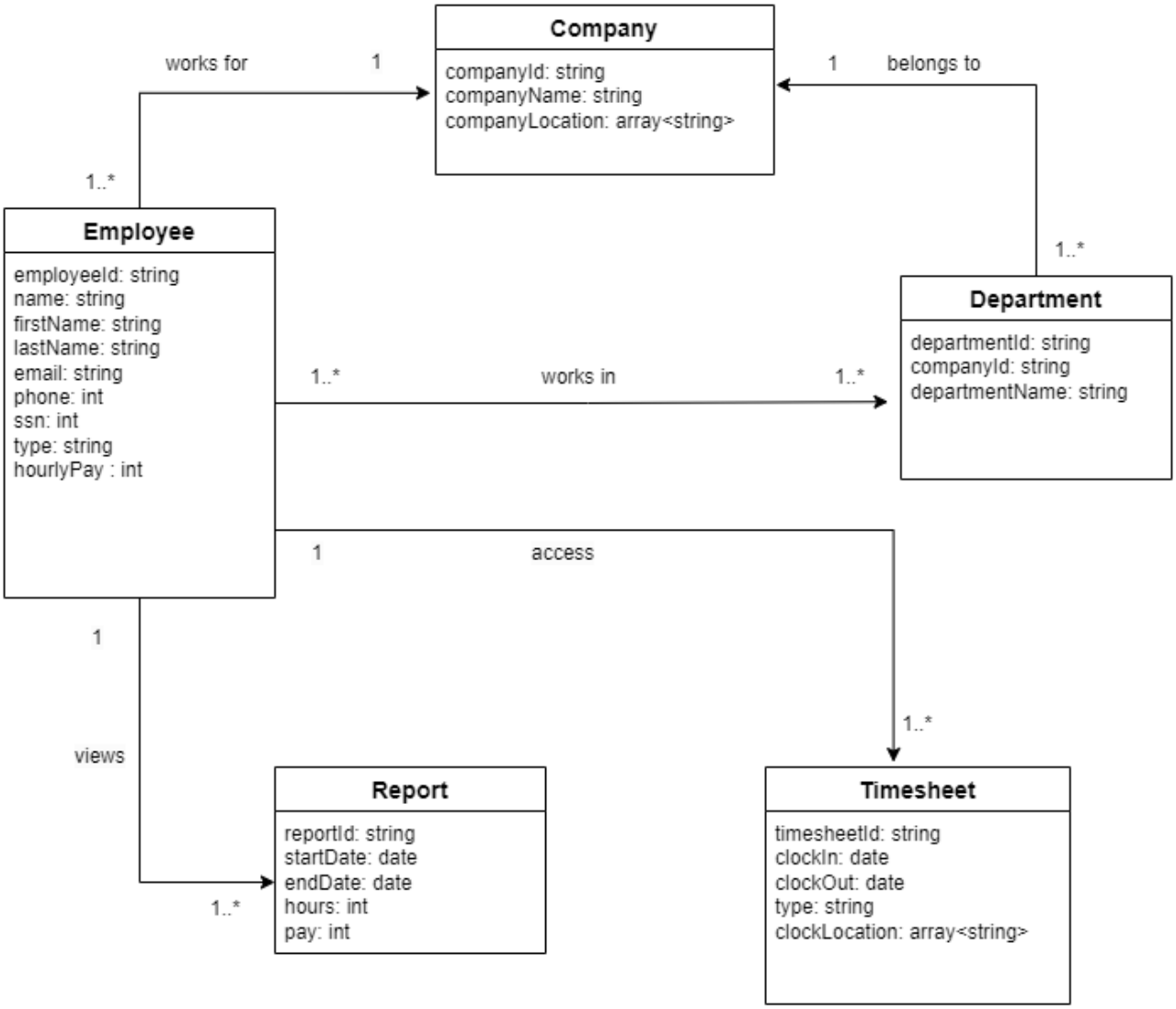
Use Case	Priority	Effort	Depends on	Assigned to	Iteration 1 (02/26/2024)	Iteration 2 (04/01/2024)	Iteration 3 (04/24/2024)
UC1	1	2	None	JG, SN	1	1	
UC2	1	2	UC1	PS, HJ	1	1	
UC3	4	1	UC2	NP			1
UC4	5	1	UC2	HJ			1
UC5	2	4	UC2	PS,JG, SN		2	2
UC6	2	4	UC2, UC5	SN,HJ, NP		2	2
UC7	3	3	UC2, UC5, UC6	PS, NP, JG		1	2
UC8	3	3	UC1, UC2, UC5, UC6, UC9	JG,HJ, SN		1	2
UC9	2	2	UC1, UC2	PS, NP		2	
UC10	4	1	UC1, UC2	SN,HJ		1	
UC11	4	1	UC1, UC2	JG, NP		1	
UC12	1	2	UC2	PS	2		
Total effort		**26**			**4**	**12**	**10**

1 person-Week = 5 hrs

Team members:

- Param Shah-**PS**, Hetvi Joshi-**HJ**, Jainik Gadara-**JG**, Nandish Patel-**NP**, Sakshi Naik-**SN**

Domain Model



Expanded Use Case

EUC1: Register Employee

Precondition: This use case assumes that the HR is logged into the system and is on the home page.	
Actor: HR	System: GeoCompensate
	0. System displays the HR home page (See Figure 1).
1. TUCBW the HR clicks the “Register Employee” button.	2. System displays the register employee form (See Figure 2).
3. The HR enters the employee’s details and clicks the “Register” button.	*4. System checks the submitted info and shows a confirmation message if no error is found and sends the credential in email (See Figure 3).
5. TUCEW the HR sees the success message.	
Postcondition: The credentials email is sent to the employee and employee is available for search.	

EUC2: Login

Precondition: The use case assumes that the employee and HR has already been registered with the system.	
Actor: HR,Employee	System: GeoCompensate
	0. System displays the login page (See Figure 4).
1. TUCBW the user clicks the “Log In” button.	*2. System verifies the entered credentials with database and redirects to the home page (See Figure 1, Figure 5).
3. TUCEW the user gets successfully logged into the system.	
Postcondition: The user is able to see the home page.	

EUC3: Edit Profile

Precondition: This use case assumes that the user has logged into the system and is on the profile page.	
Actor: HR, Employee	**System:** GeoCompensate
	0. System displays the Profile page. (See Figure 6)
1. TUCBW the user clicks the “Edit profile” button.	*2. System checks the submitted info and displays the “Save Changes” button. (See Figure 7).
3. The user clicks the “Save Changes” button.	4. System verifies the changed fields and shows a success message (See Figure 8).
5. TUCEW the user sees the “Profile has been updated successfully” message.	
Postcondition: The user can see the updated information in the profile page and updates the submitted information inside the database.	

EUC4: Change Password

Precondition: This use case assumes that the user has logged into the system and is on the profile page.	
Actor: HR, Employee	**System:** GeoCompensate
	0. System displays the Profile page with a “Change Password” button. (See Figure 6)
1. TUCBW the user selects the “Change password” button.	2. System displays the change password form (See Figure 9).
3. The user enters the required credentials in the form and click on “Update Password” button.	*4. System verifies the changed fields and shows a success message (See Figure 10).
5. TUCEW the user sees the “Password has been changed successfully” message.	
Postcondition: The employee(s) can now login with a new password into the system.	

EUC5: Clock In

Precondition: This use case assumes that the Employee has logged into the system and is on the home page.	
Actor: Employee	**System:** GeoCompensate
	0. System displays the Home page (See Figure 5).
1. TUCBW the employee clicks the “Clock In” button.	*2. System checks the geolocation of the employee and clocks in the employee (See Figure 11).
3. TUCEW the employee sees the “Clocked In” message.	
Postcondition: The employee can see the clock in time on the home screen and can now clock out.	

EUC6: Clock out

Precondition: This use case assumes that the Employee has logged into the system, has already clocked in and is on the home page.	
Actor: Employee	**System:** GeoCompensate
	0. System displays the Home page (See Figure11).
1. TUCBW the employee clicks the “Clock Out” button.	*2. System checks the employee's clock in data and clocks out (See Figure 5).
3. TUCEW the employee sees the “Clocked out” message.	
Postcondition: The employee can see the clock out time on the home screen and can now clock in again.	

EUC7: View Timesheet

Precondition: This use case assumes that the Employee has logged into the system and is on the home page.	
Actor: Employee	**System:** GeoCompensate
	0. System displays the home page (See Figure 5).
1. TUCBW the employee clicks the “Timesheet” button.	*2. System redirects the employee to the timesheet page and displays data from the employee timesheet of the month (See Figure 12).
3. TUCEW the employee sees monthly working hour details.	
Postcondition: The employee sees timesheet records which includes date, total hours worked and total wage for a day for the last 30 days.	

EUC8: View Employee’s Timesheet

Precondition: This use case assumes that the HR has logged into the system and is on the search employee page with searched employee data.	
Actor: HR	**System:** GeoCompensate
	0. System displays the Search Employee page table grid (See Figure 13).
1. TUCBW the HR clicks the “View Timesheet”.	*2. System checks the submitted info and displays the searched employee timesheet records (See Figure 14).
3. TUCEW the HR sees the employee's clock-in/clock-out details day wise for that month.	
Postcondition: HR can see the employee’s timesheet with each clock ins and clock outs in detail.	

EUC9: Search Employee

Precondition: This use case assumes that the HR has logged into the system and is on the search employee page.	
Actor: HR	**System:** GeoCompensate
	0. System displays the Search Employee page.(See Figure 13)
1. TUCBW the HR clicks on the search button after entering the employee first or last name.	*2. System checks the submitted info and displays the searched employee(s) records (See Figure 14).
3. TUCEW the HR sees the employee(s) records in a table format.	
Postcondition: The searched employee(s) records are available for edit profile, view timesheet and remove employee.	

EUC10: Update Employee

Precondition: This use case assumes that the HR has logged into the system, is on the search page and searched for an employee.	
Actor: HR	**System:** GeoCompensate
	0. System displays the search Employee page table grid (See Figure 13).
1. TUCBW the HR updates the required fields and clicks on the “Update Employee” button.	*2. System checks the submitted info and updates the searched employee details and displays the success message (See Figure 15).
5. TUCEW HR sees the “Employee profile updated successfully” message.	
Postcondition: The employee’s updated details are updated in the database and the employee is available for search.	

EUC11: Remove Employee

Precondition: This use case assumes that the HR has logged into the system and has already searched for the valid employee.	
Actor: HR	**System:** GeoCompensate
	0. System displays the searched employee records (See Figure 13).
1. TUCBW the HR clicks the “Relieve Employee” button.	*2. System deletes the employee data from the database and shows the message (See Figure 16,See Figure 17).
3. TUCEW the HR sees the “Relieved Employee Successfully” message.	
Postcondition: The deleted employee will not be able to search in the system.	

EUC12: Logout

Precondition: This use case assumes that the user has logged into the system.	
Actor: HR, Employee	System: GeoCompensate
	0. System displays the home page (See Figure 1 , Figure 5).
1. TUCBW the user clicks the “Logout” button.	*2. System displays the confirmation pop up (See Figure 18).
3. TUCEW the user clicks the “Yes” button and is logged out of the system.	
Postcondition: The user is redirected to the login page.	

UIPs

Fig 1.HR Home Page

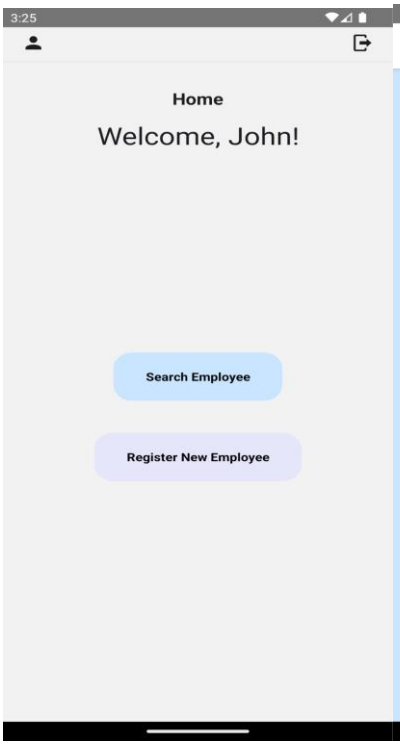


Fig 2. Register Employee Form

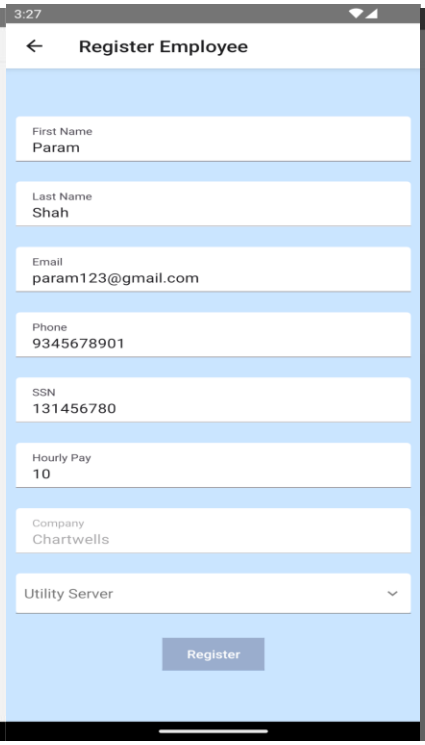


Fig 3. Success Message

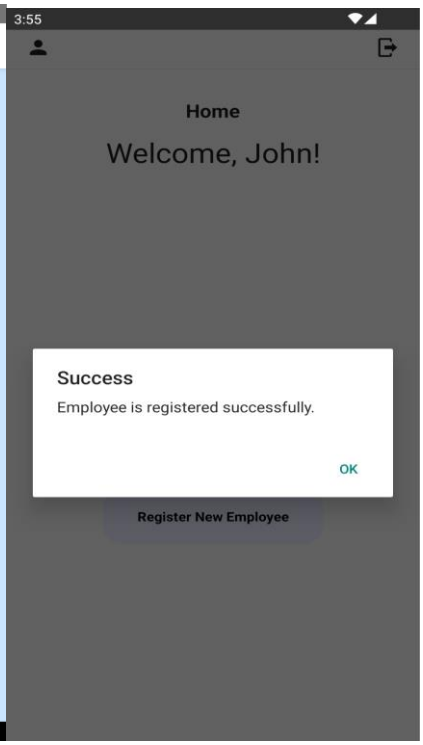


Fig 4.Login Page

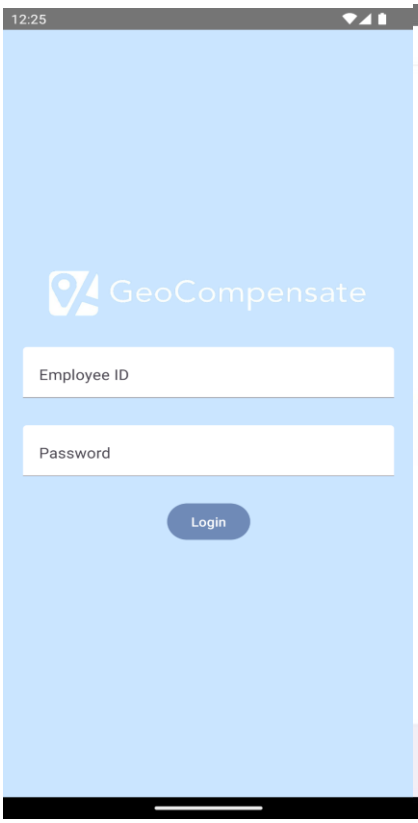


Fig 5. Employee Home Page

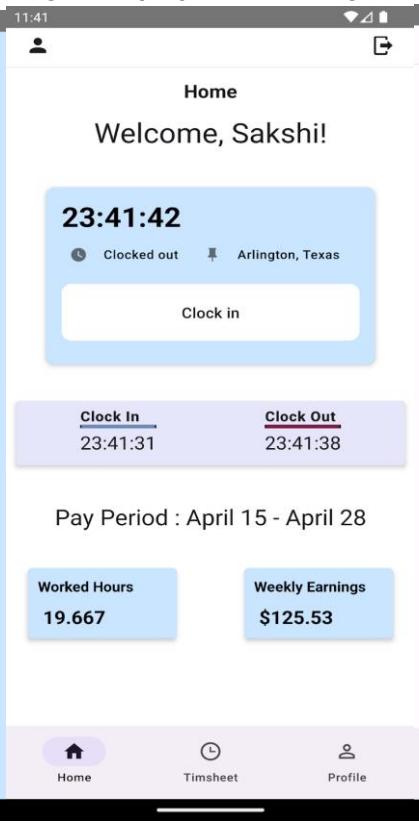


Fig 6. Employee Profile Page

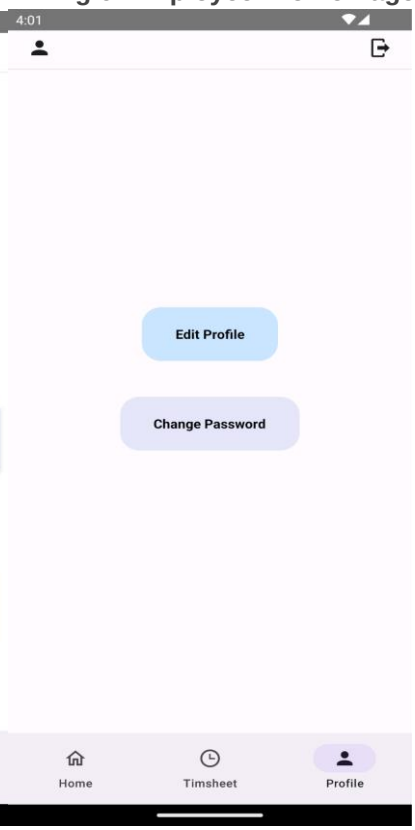


Fig 7. Save Changes

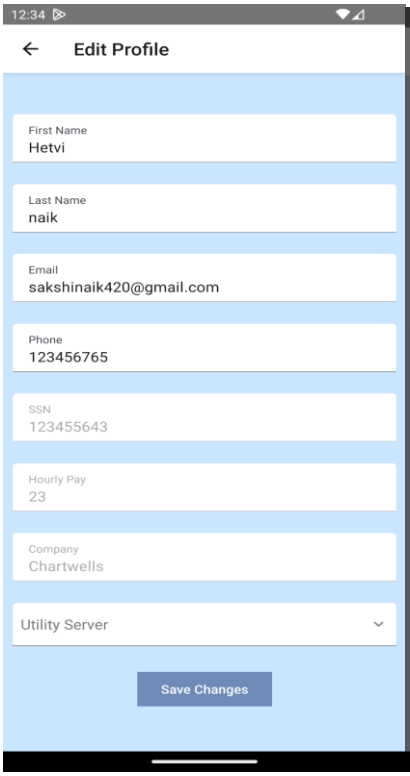


Fig 8. Success Message

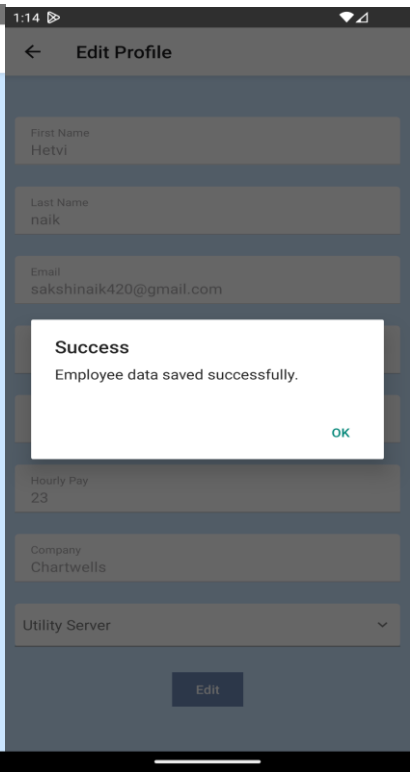


Fig 9. Change Password Form

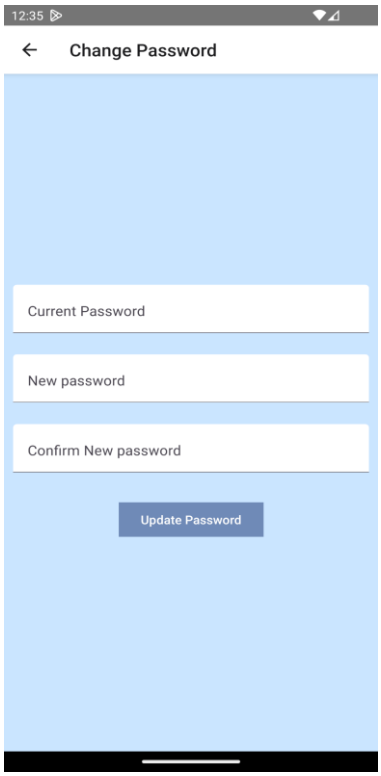


Fig 10. Change Pwd Success

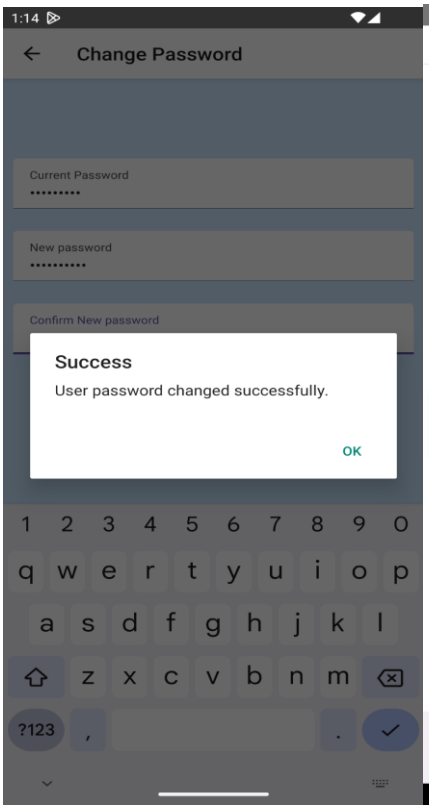


Fig 11. Clock In-Clock Out Menu

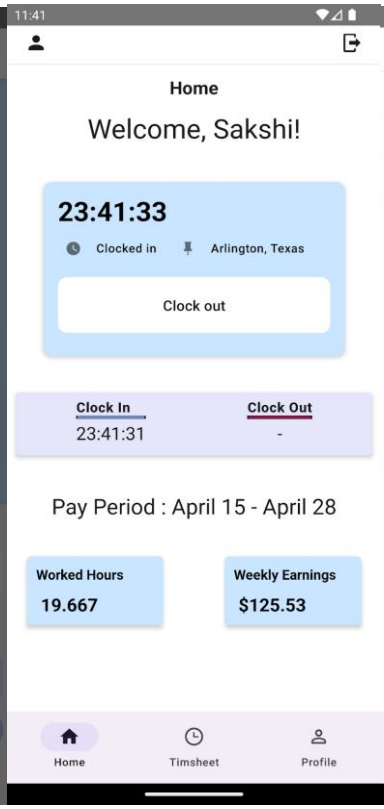


Fig 12. Timesheet Page

Fig 13.Search Employee Page

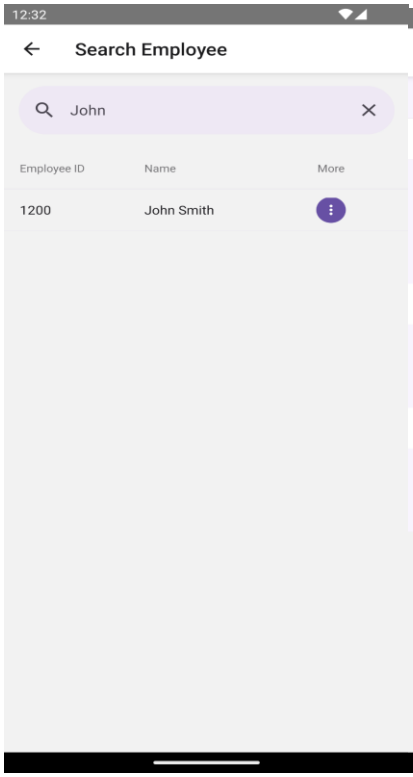


Fig 14. Emp's Timesheet

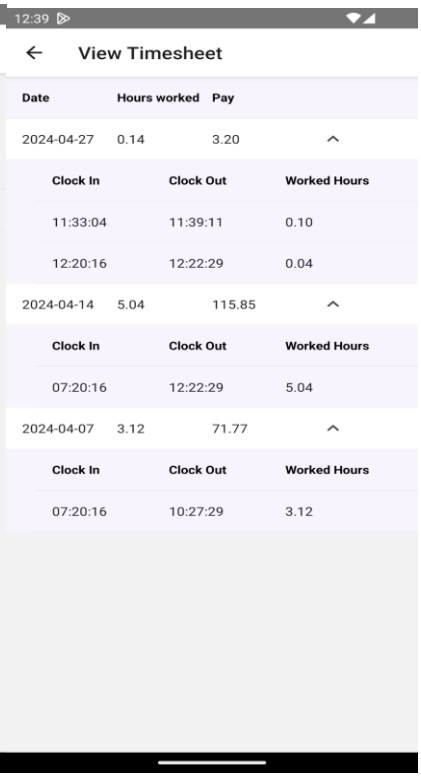


Fig 15. Update Emp Success

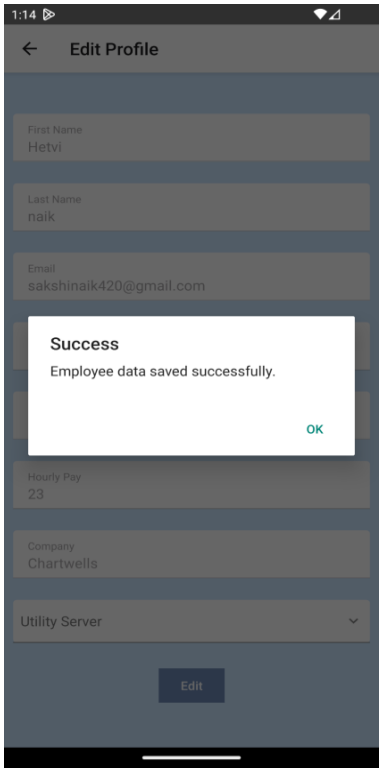


Fig 16.Remove Emp Confirmation

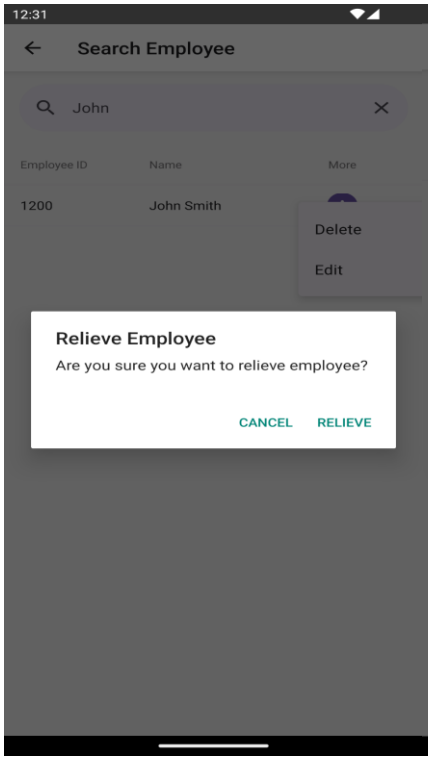


Fig 17.Relieved Emp Message

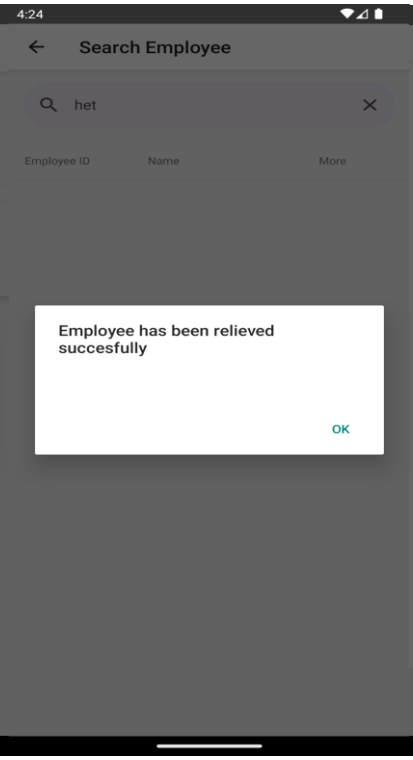
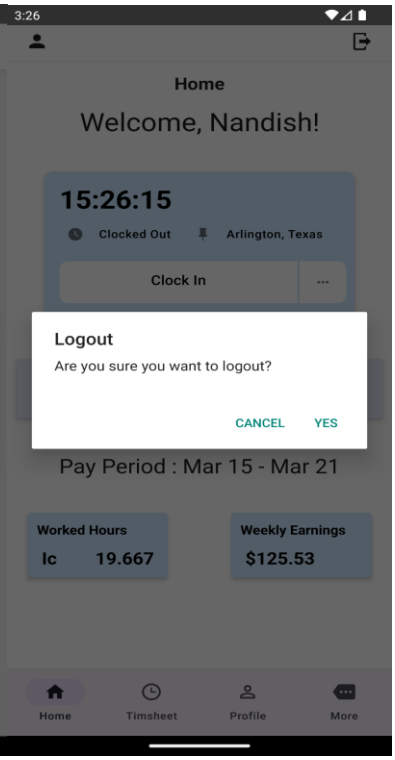


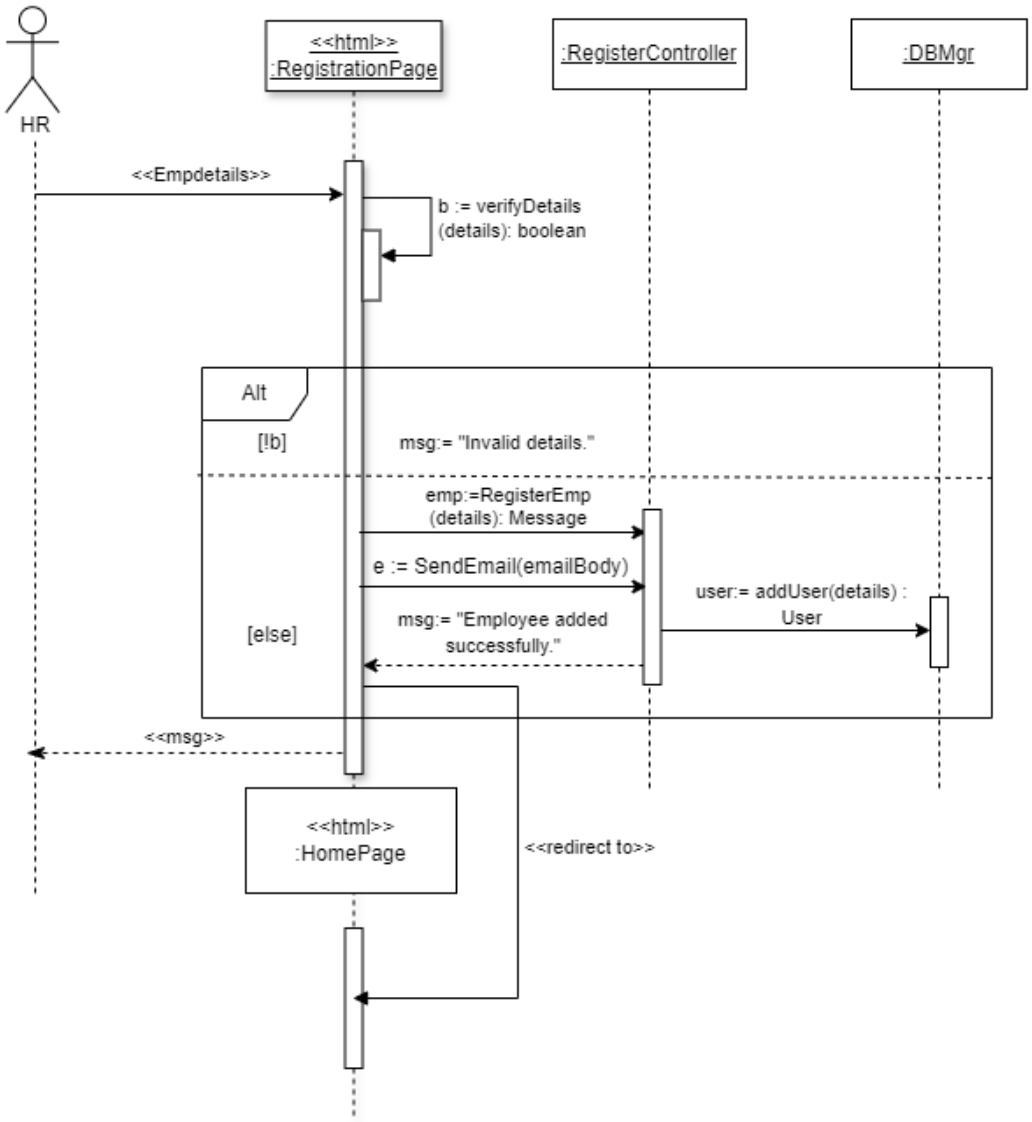
Fig 18.Logout Confirmation



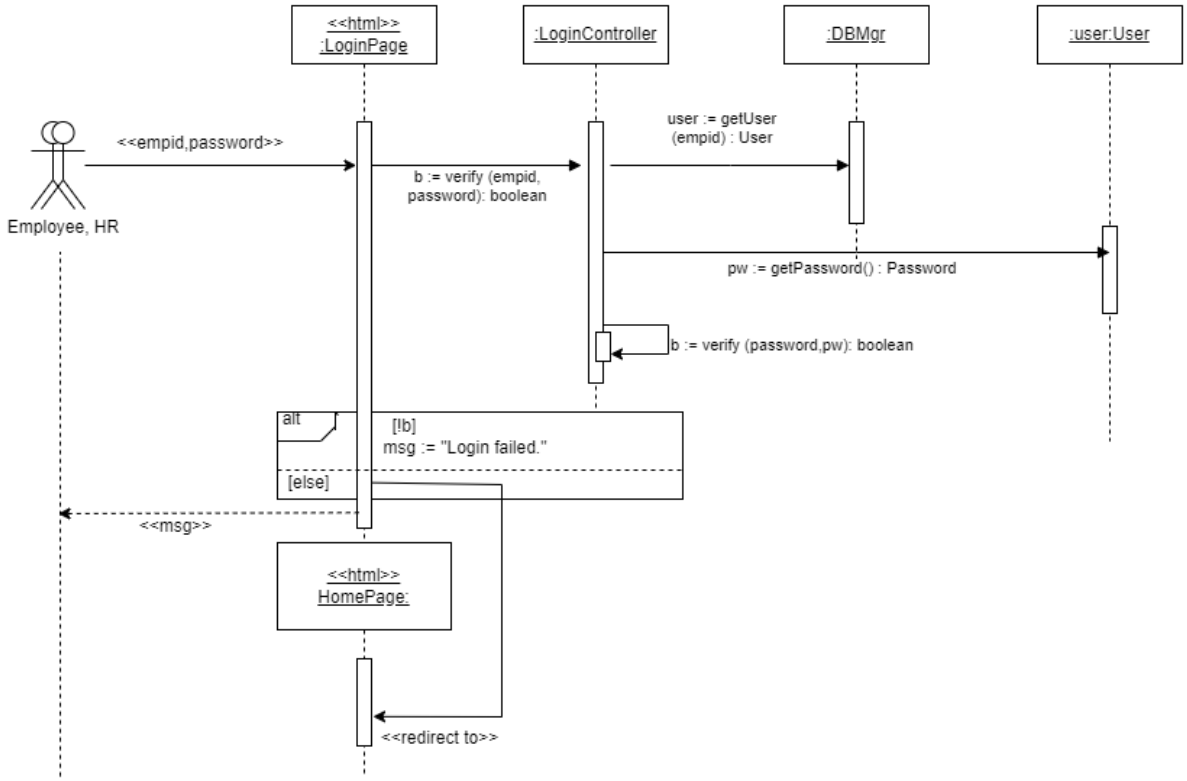
.

Sequence Diagram

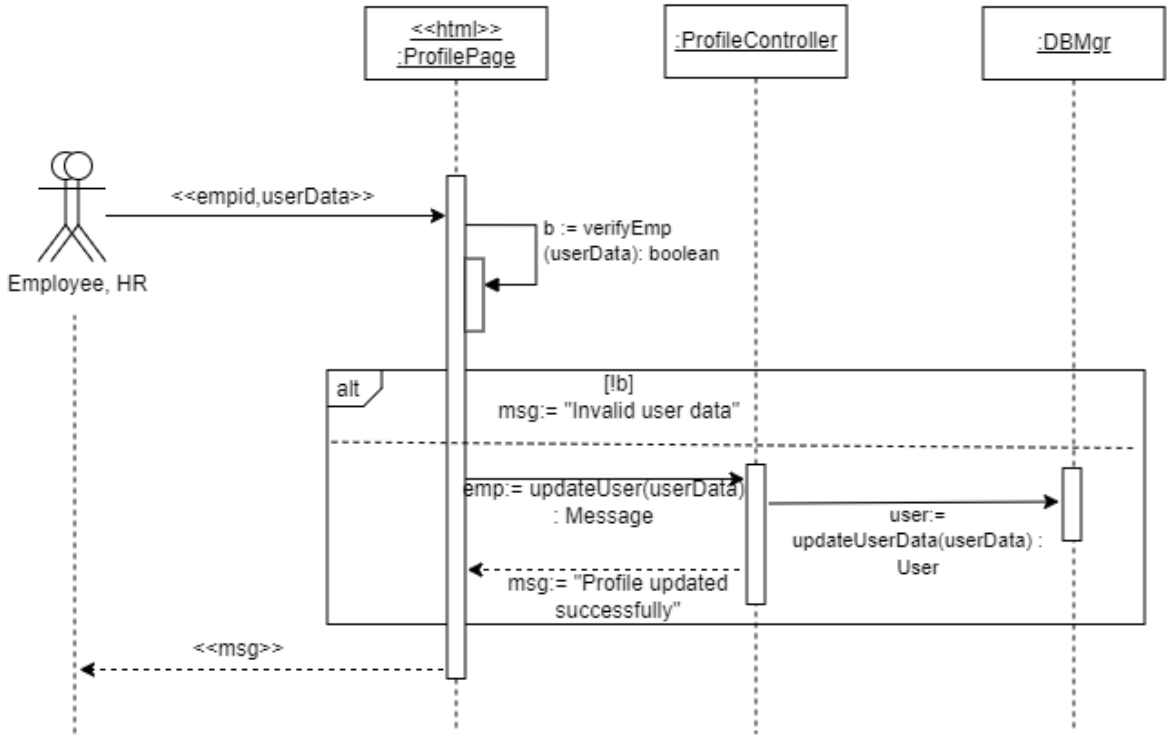
1. Register Employee Sequence Diagram



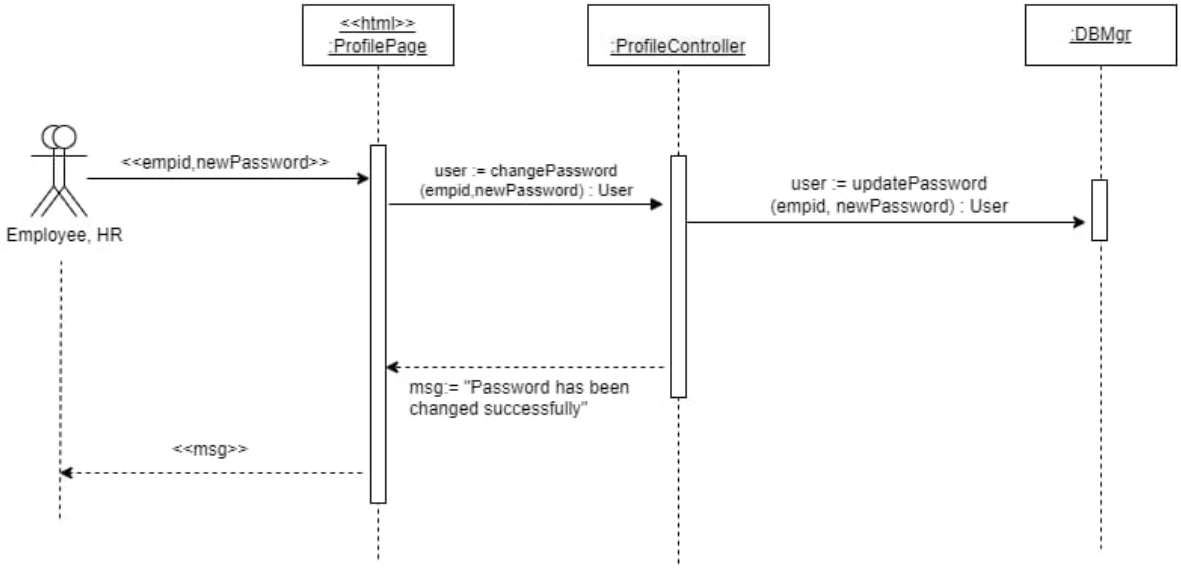
2. Login Sequence Diagram



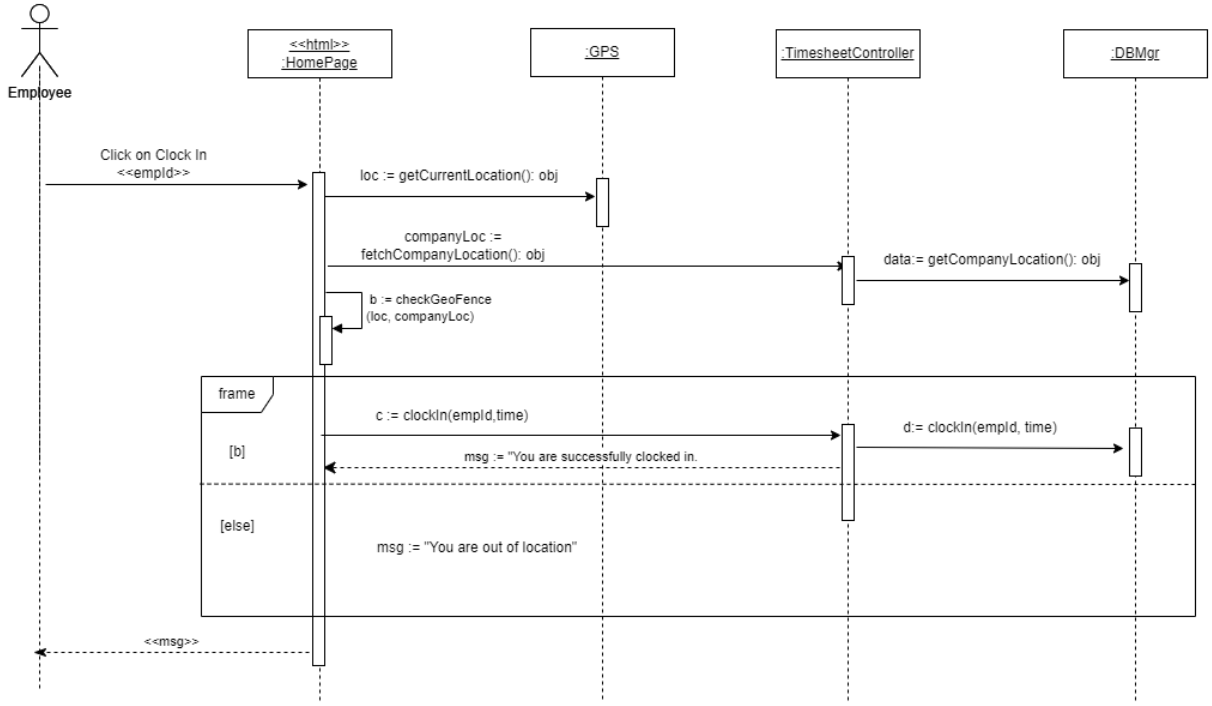
3. Edit Profile Sequence Diagram



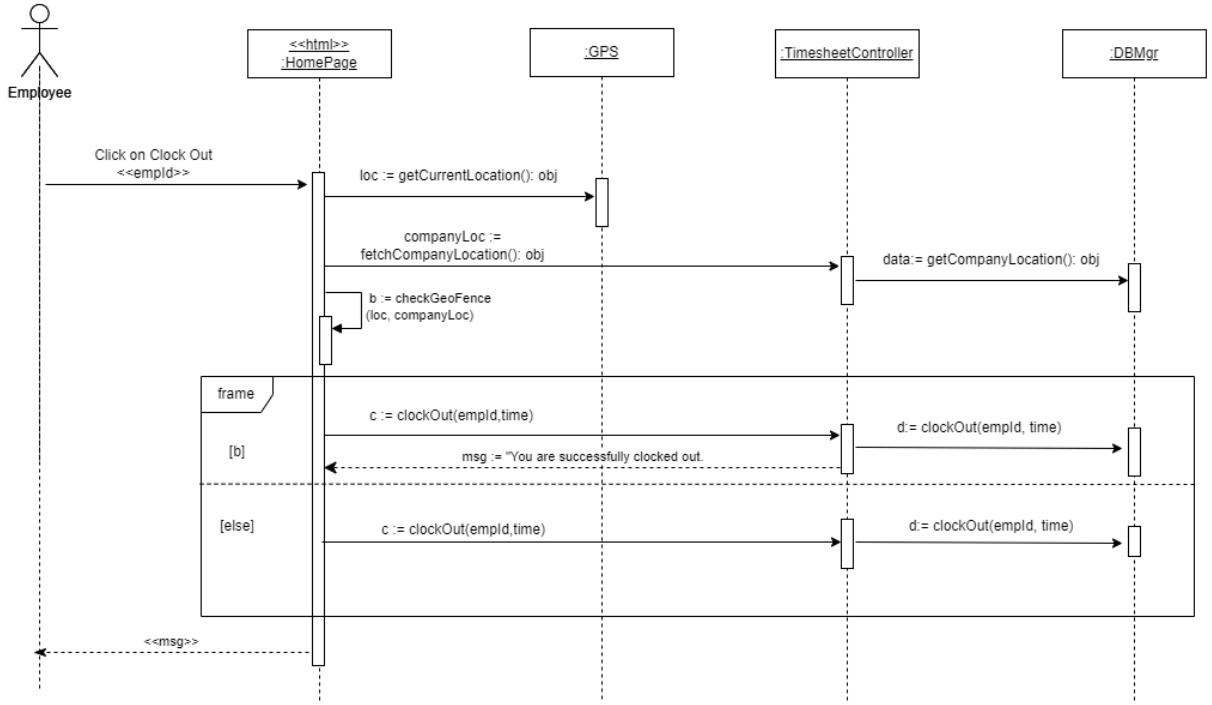
4. Change Password Sequence Diagram



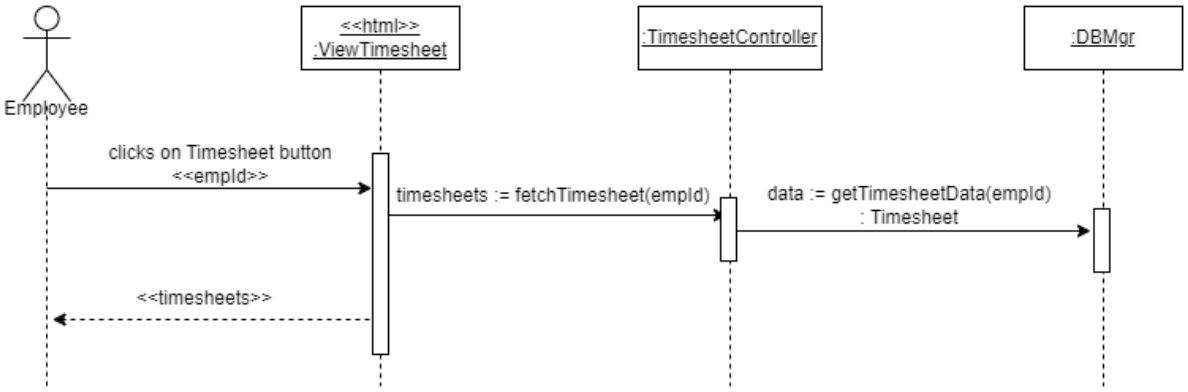
5. Clock In Sequence Diagram



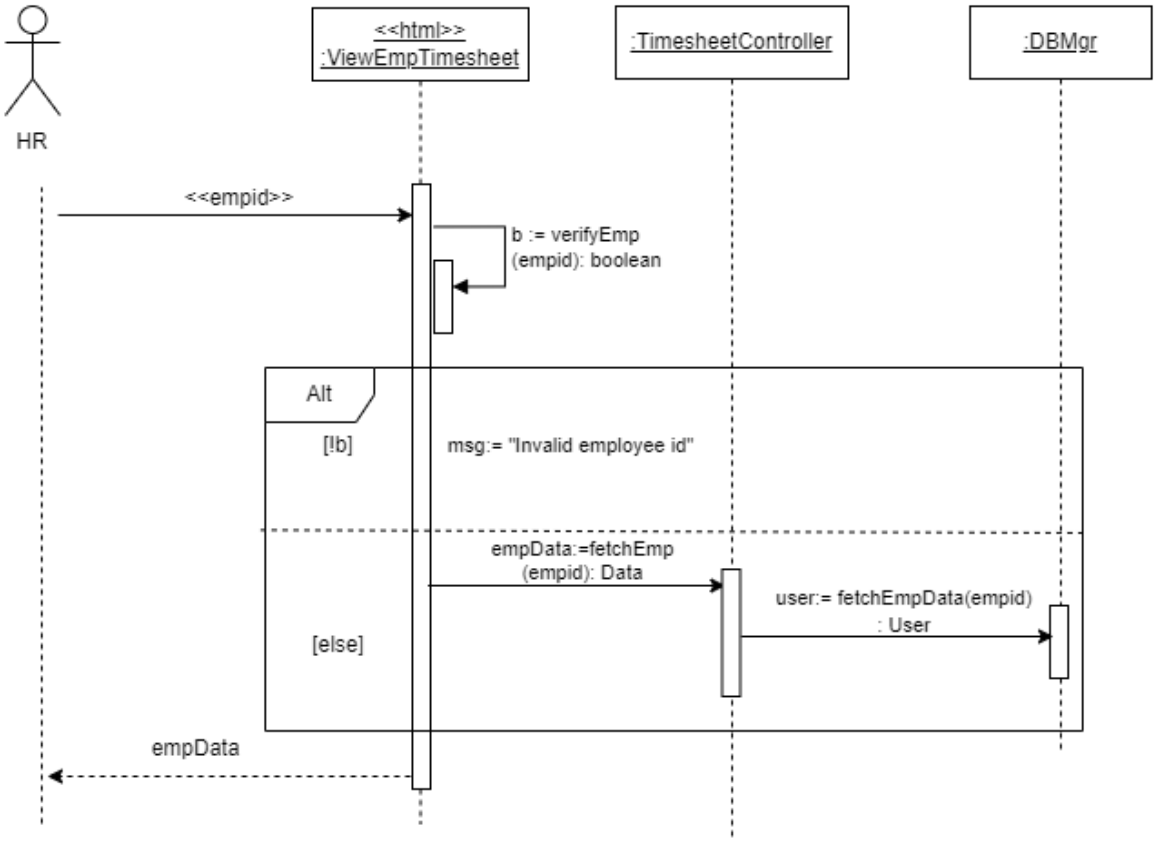
6. Clock Out Sequence Diagram



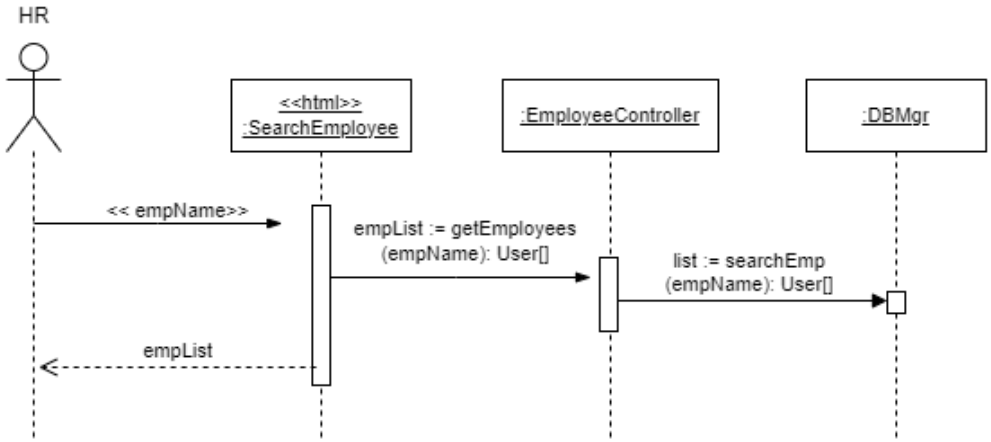
7. View Timesheet Sequence Diagram



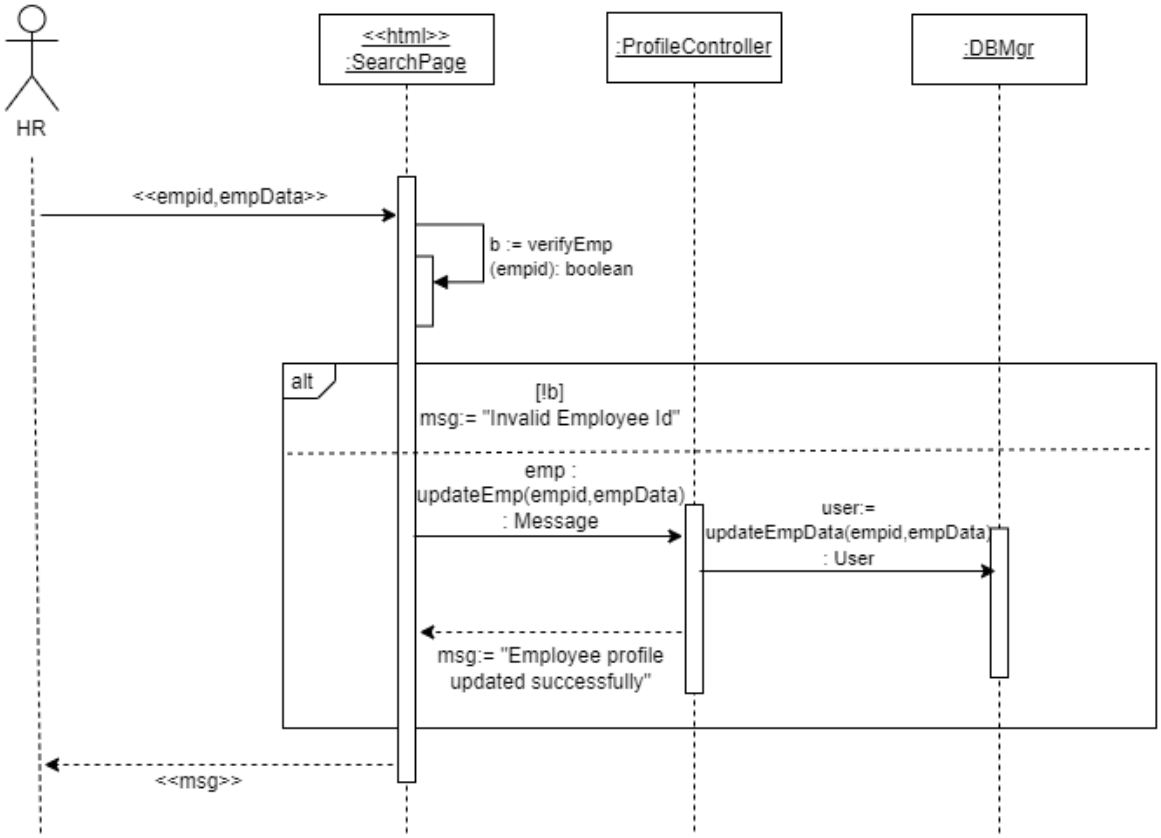
8. View Employee's Timesheet Sequence Diagram



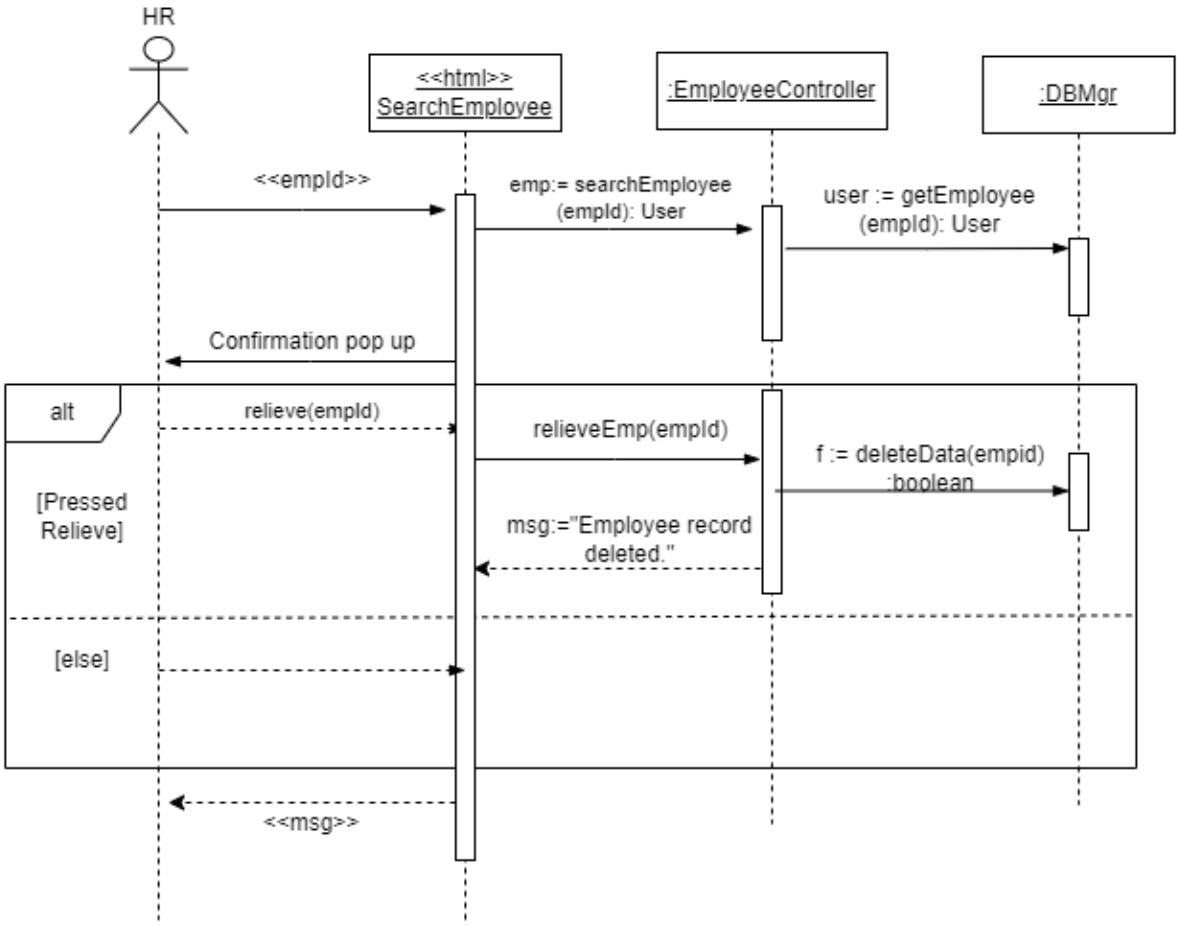
9. Search Employee Sequence Diagram



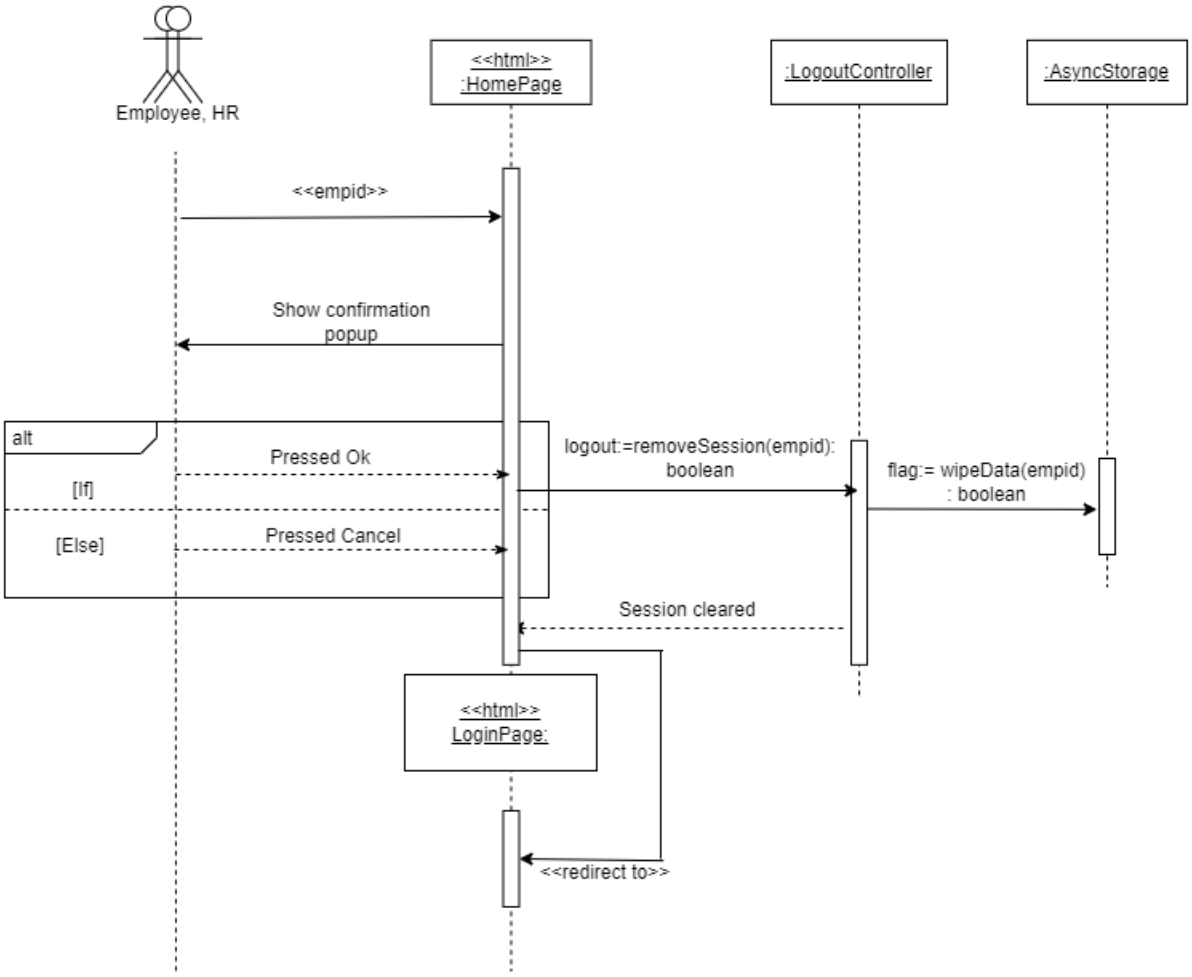
10. Update Employee Sequence Diagram



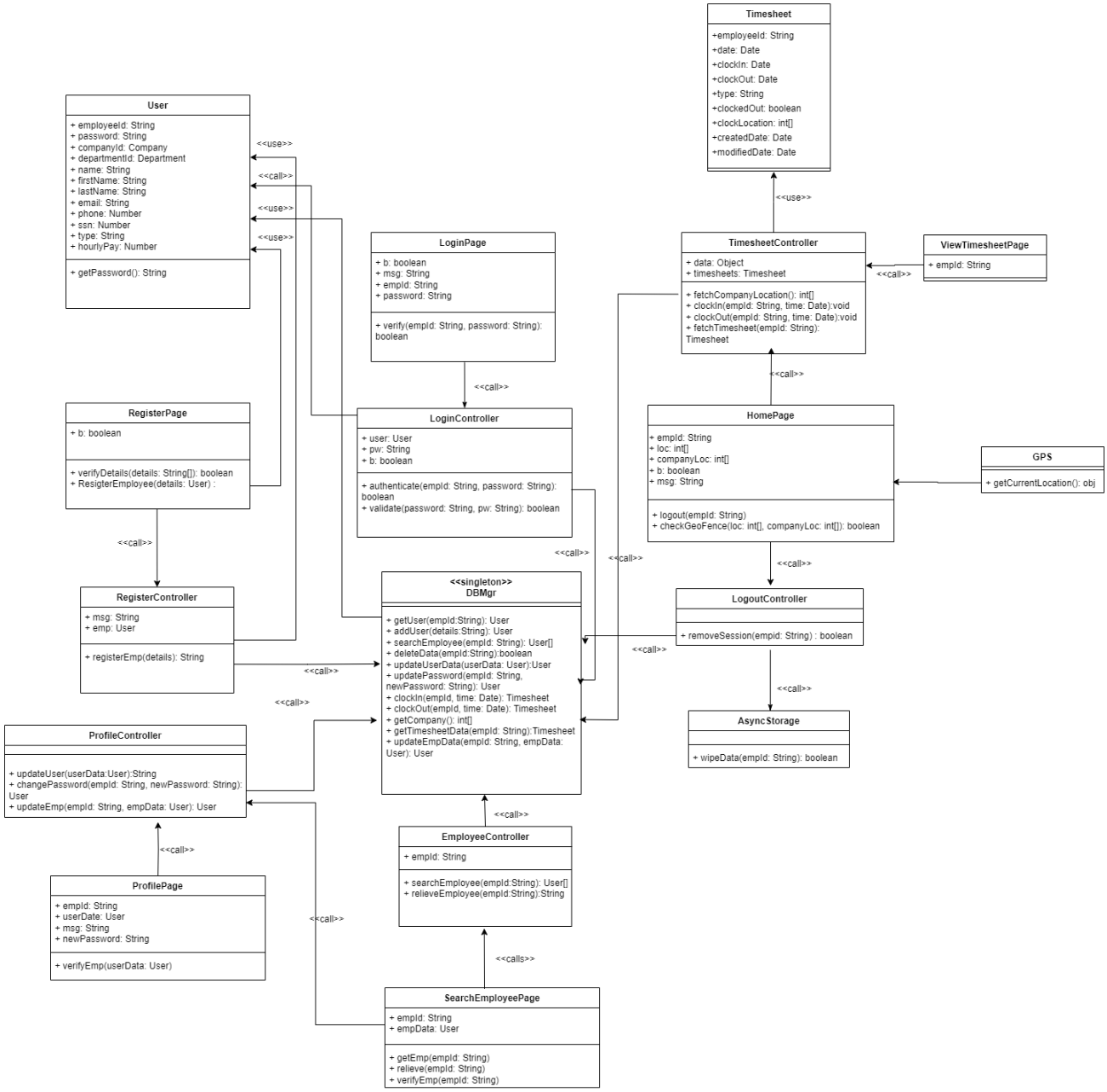
11. Remove Employee Sequence Diagram



12. Logout Sequence Diagram



Design Class Diagram



App Demo

Link: https://youtu.be/GqKjJTQcdX8