**Analysis Document**

Orin Gavriel 318774338

Or Goren 314619115

Yehonatan Mekayten 213637424

**System Screens Structure**

**List of screens:**

General pages:

* Login page
* Registration page
* Forgot password page

Employee screens:

* Employee home page
* Employee menu
* Employee My profile page
* Employee work arrangement page
* Employee submitting constraints page
* Employee submitting day off request page
* Employee submitting shift exchange request page
* Employee requests status page
* Employee work updates page

manager screens:

* Manager home page
* Manager menu
* Manager My profile page
* Manager employees requests page
* Manager build work arrangement page
* Manager work arrangement published page
* Manager send updates page
* Manager sent updates page

**Connections Between Screens:**

Login page:

* When an employee enters his username and password and clicks the “Login” button he will go to the Employee home page.
* When a manager enters his username and password and clicks the “Login” button he will go to the Manager home page.
* Clicking the “Create account” button leads to the Registration page
* Clicking the “Forgot password” button  leads to the Forgot password page

Employee home page:

* Clicking “הפרופיל שלי” button in the menu leads to My profile page.
* Clicking “סידור עבודה” button in the menu leads to Employee work arrangement page.
* Clicking “הגשת אילוצים” button in the menu leads to Employee submitting constraints page.
* Clicking “בקשת יום חופש/מחלה” button in the menu leads to Employee submitting day off page.
* Clicking “בקשת החלפת משמרת” button in the menu leads to Employee submitting shift change request page.
* Clicking “סטטוס בקשות” button in the menu leads to Employee requests status page.
* Clicking  “עדכונים והודעות” button in the menu leads to Employee notifications page.

Manager home page:

* Clicking “הפרופיל שלי” button in the menu leads to My profile page.
* Clicking “בקשות עובדים” button in the menu leads to Manager employees requests page.
* Clicking “ בניית סידור עבודה” button in the menu leads to Manager publish work arrangement page.
* Clicking “עדכונים לעובדים” button will open sub menu with two options- “שליחת עדכון”

and “עדכונים שנשלחו”. Clicking “שליחת עדכון” button in the menu leads to Manager send notifications page and Clicking “עדכונים שנשלחו” button in the menu leads to Manager sent notifications page.

**Visual representation**

powerpoint presentation is added

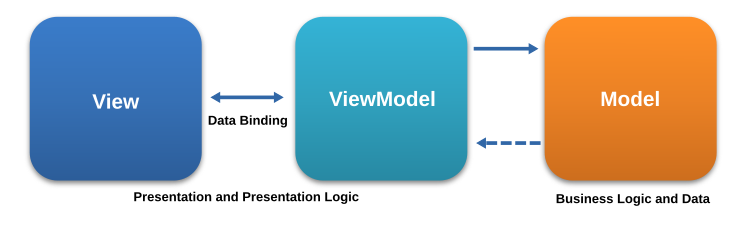
**Design Patterns for Android Studio**

we intend to use MVVM architecture in our project.

MVVM (Model-View-ViewModel) is a software architectural pattern that separates the user interface from the business logic. It consists of three key components:

1. Model: Manages the application's data and business logic.
2. View: Represents the user interface. It is responsible for defining the structure, layout and appearance of the user's screen and does not contain any business logic.
3. ViewModel: links between the View and the Model.  if any change occurs it notifies the view by data binding . the viewmodel implements properties and commands to which the view can data bind to, and notifies the view of any state changes.

MVVM helps make the code more maintainable, testable, and scalable by separating concerns and ensuring each part has a clear responsibility.



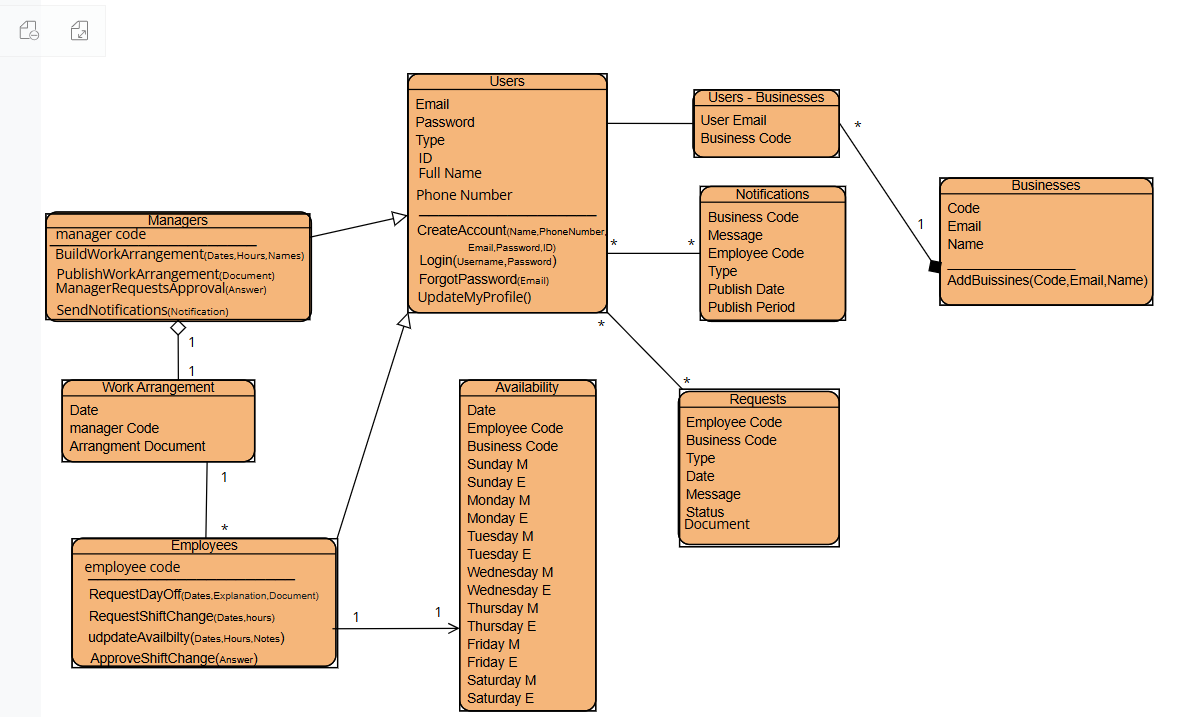
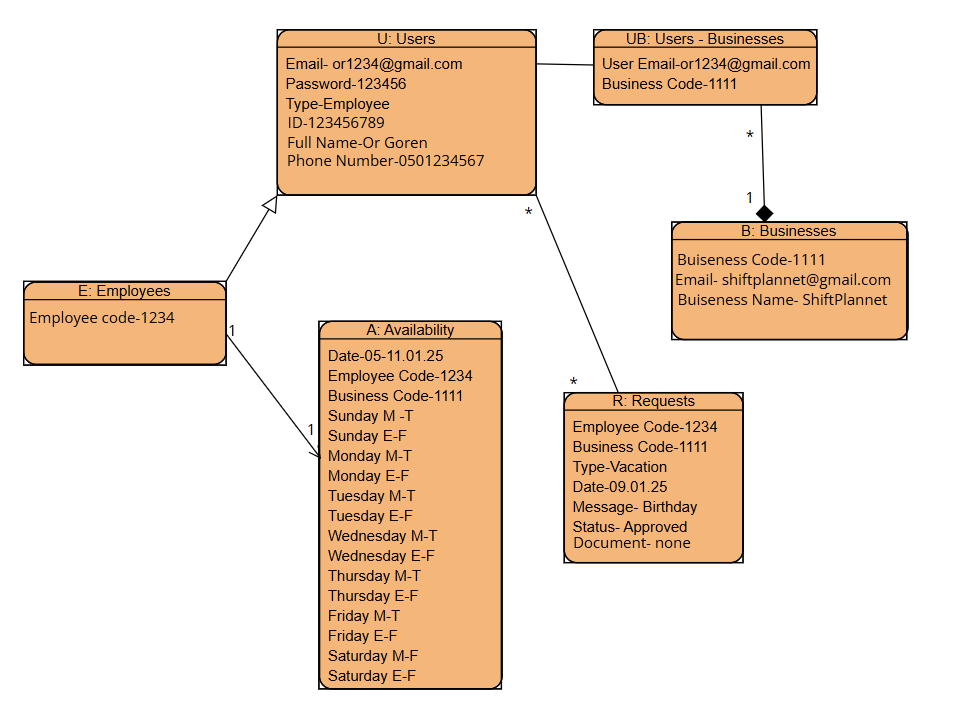
Our app, ShiftPlannet, involves dynamic user interactions, real-time updates, and complex business logic and due to this reasons the most suitable architectural pattern is MVVM.

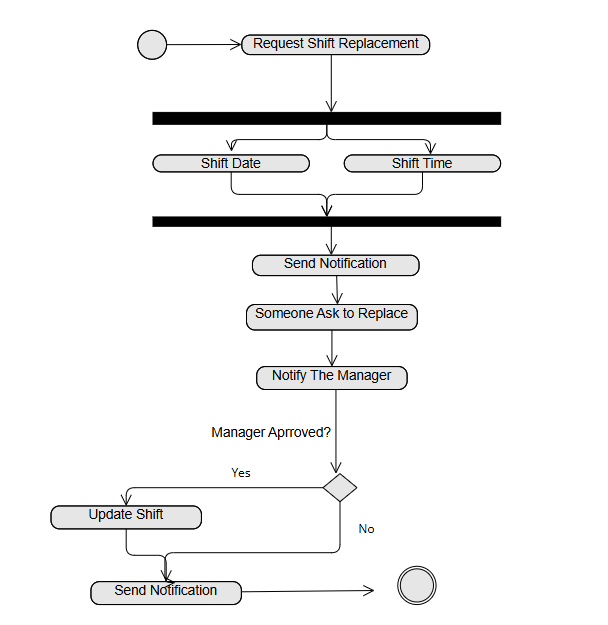
Using MVVM offers several significant benefits for maintainability, scalability, and performance as followed:

1. Separation of components: MVVM clearly separates the logic of managing data (ViewModel), presenting data (View), and handling business logic (Model). updates to the UI or changes in business logic can be done independently, making the app more modular and easier to maintain.
2. Scalability: MVVM supports scalability by allowing independent development of the UI and logic. As the application grows, new features can be added and modifications can be done without affecting other parts of the system. This flexibility will allow the app to grow and change without significant refactoring and also keep a clean code.
3. Data Binding: MVVM allows automatic data binding between the View and ViewModel, reducing the amount of code needed to update the UI. This results in cleaner code, better performance and smoother user experience.
4. Testability: the ViewModel handles the business logic separately from the UI (View) and data (Model), making it easier to test the logic without involving the UI. As a result, unit tests can focus on the core functionality of the app ensuring better quality and reducing bugs. this will make the testing process more efficient and reliable.

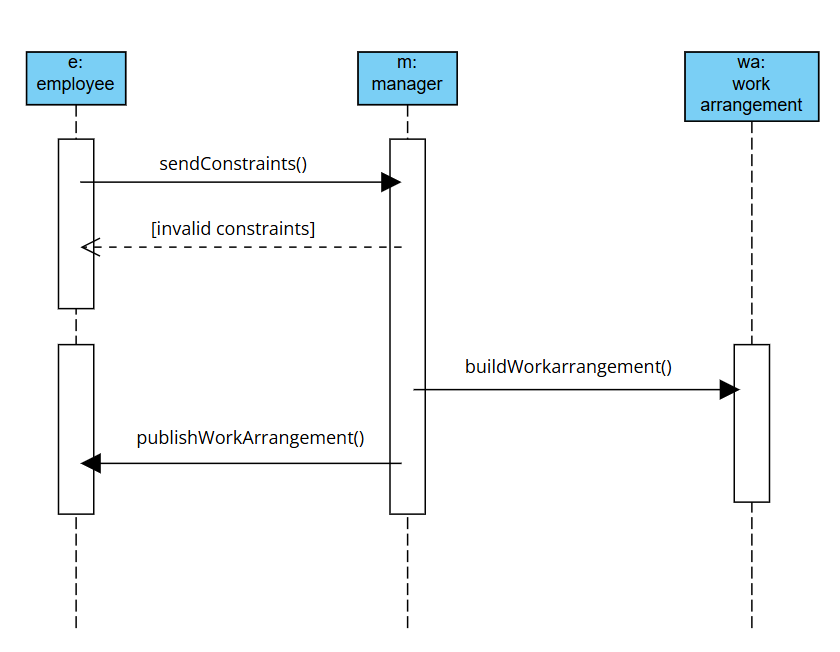
Overall, MVVM will help the app stay flexible, scalable, and maintainable as it continues to grow and develop.

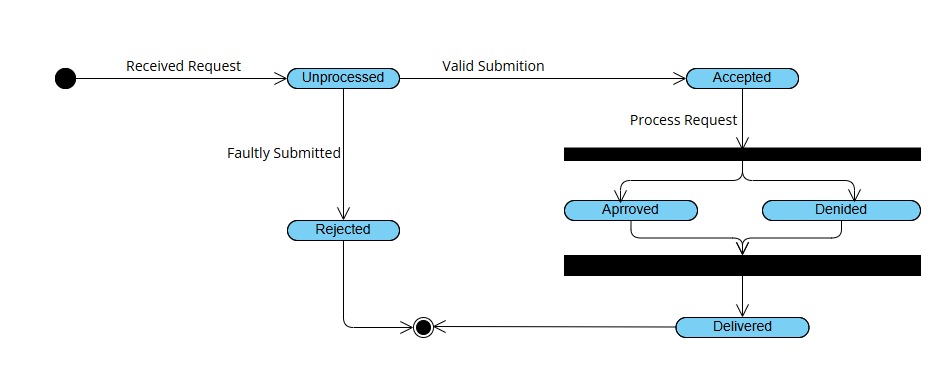
**Diagrams**

* Class Diagram
* Object Diagram
* Activity Diagram – request shift replacement.



* Sequence Diagram – build and publish work arrangement.



* State Machine Diagram – day off request
* ERD (Entity-Relationship Diagram)

