

# Design Deliverables

## Domain Model

### 10 Potential Nouns

1. Users
2. Groups
3. Locations
4. Itinerary
5. Transportation
6. Username
7. ArrivalTime
8. DepartTime
9. listOfUsers
10. Vehicle

### Classes and Attributes

#### **Users**

username, password

#### **Itinerary**

startDate, endDate, listOfLocs

#### **Destinations**

locName, city, country

#### **Transportation**

vehicle, departTime, arrivalTime

#### **Lodging information**

hotelName, checkIn, checkOut

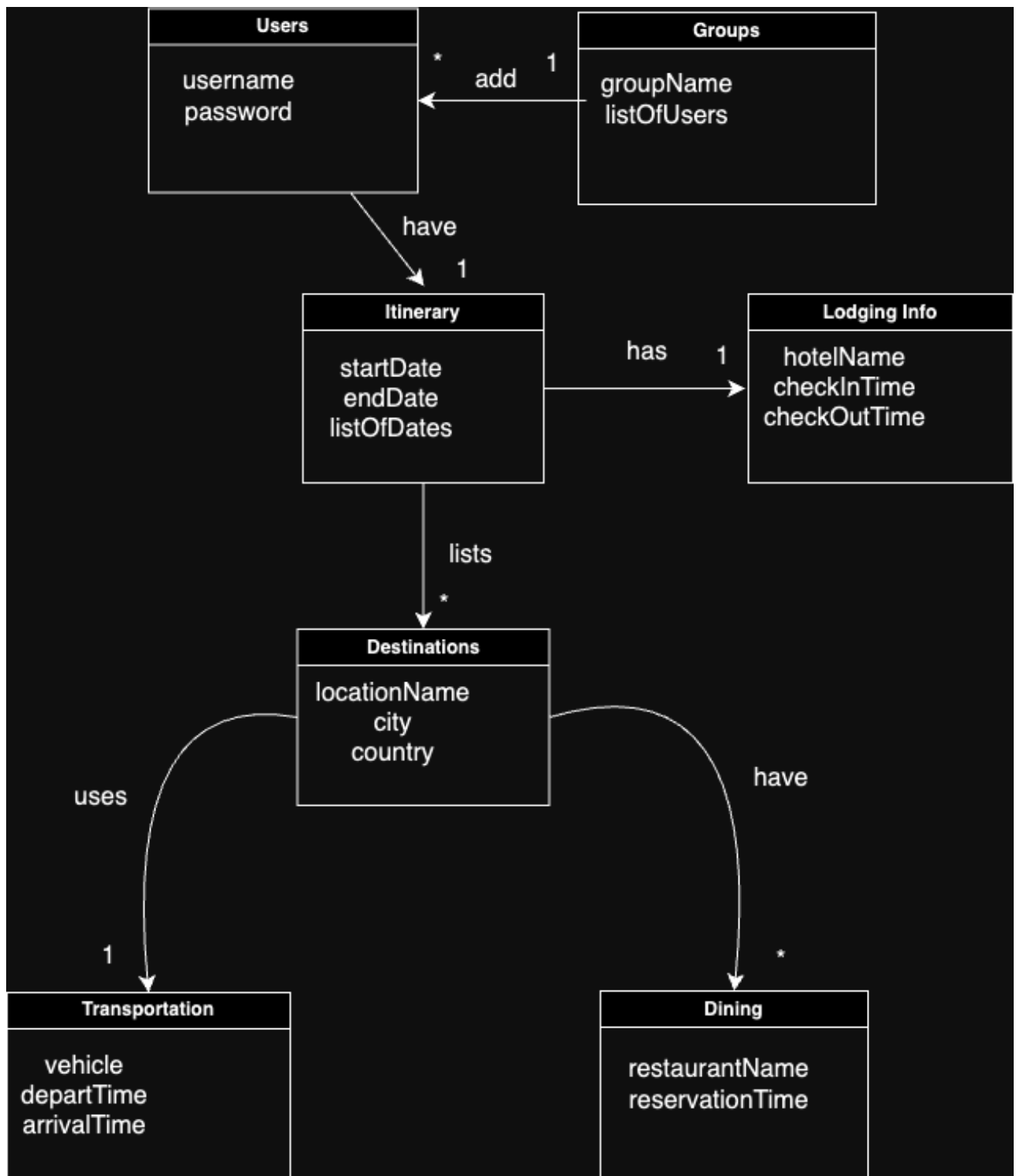
#### **Dining**

restaurantName, reservationTime

#### **Groups**

groupName, listOfUsers

## Domain Diagram



# Use Case Diagram

## Primary Actors

User, Travel Agent

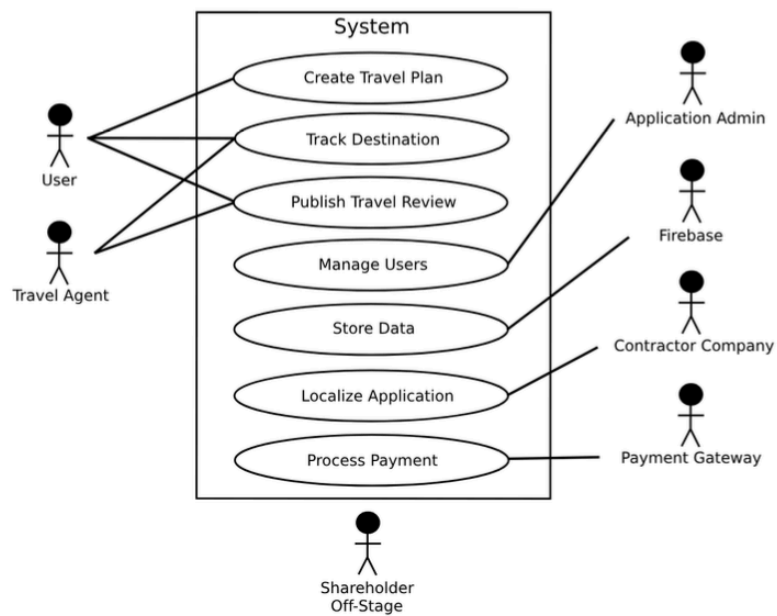
## Supporting Actors

Application Admin, Third-party database service (Firebase), Contractor Company

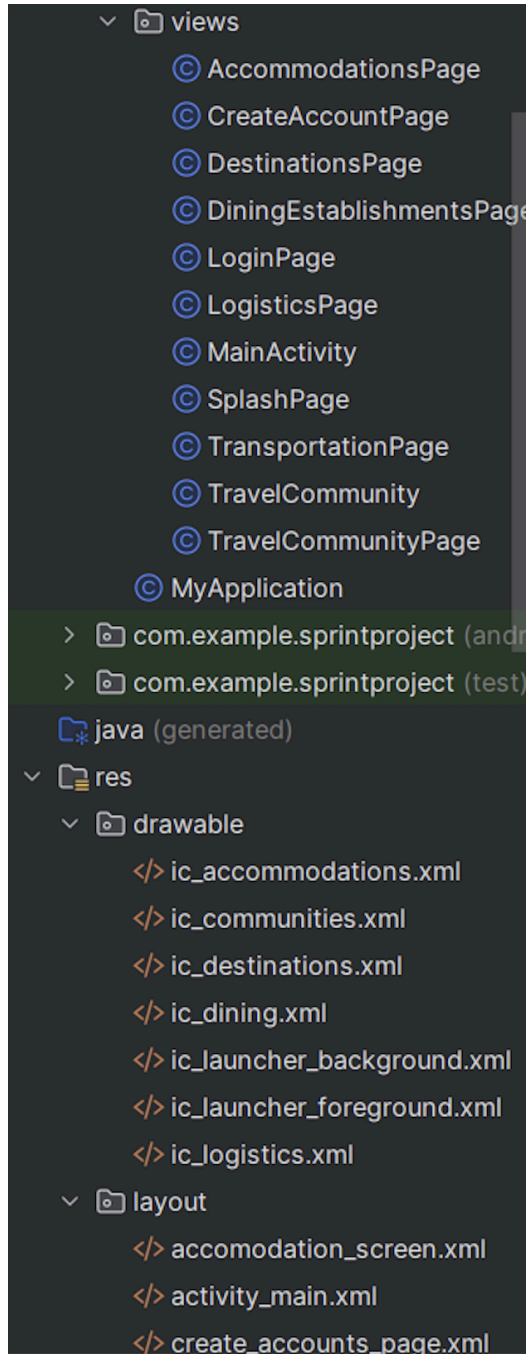
## Off-Stage Actors

Shareholder

**Additional Actor:** Payment Gateway (Supporting)



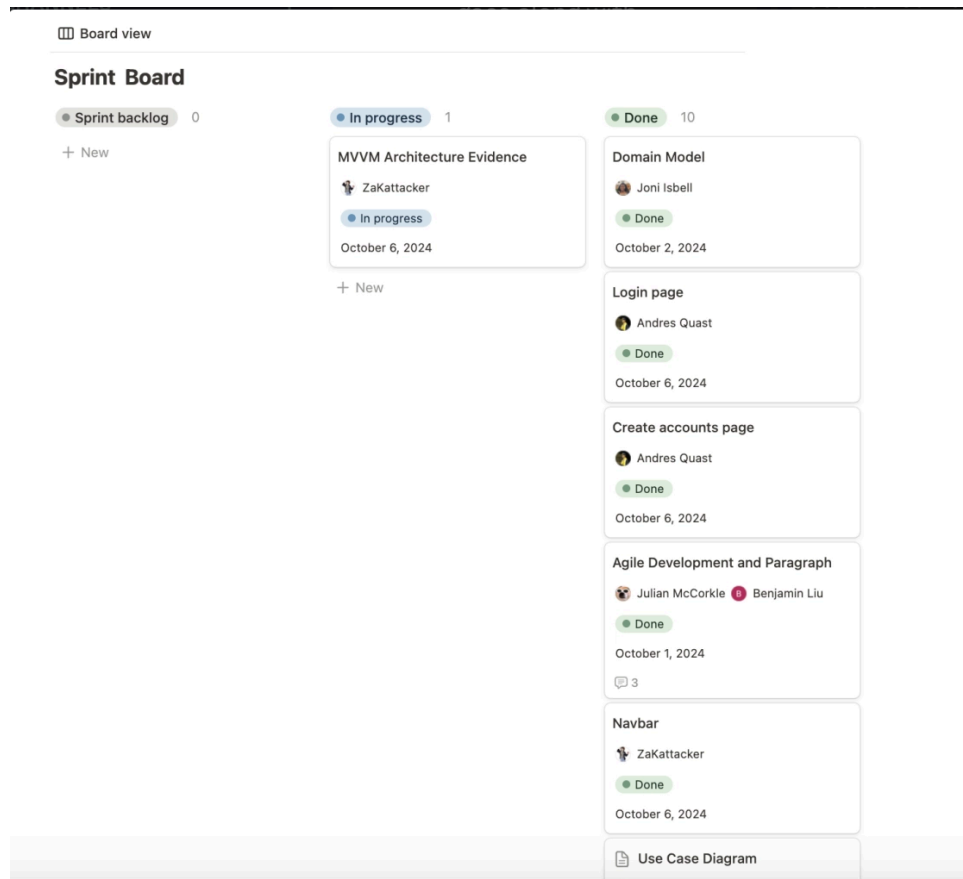
## MVVM Architecture Evidence

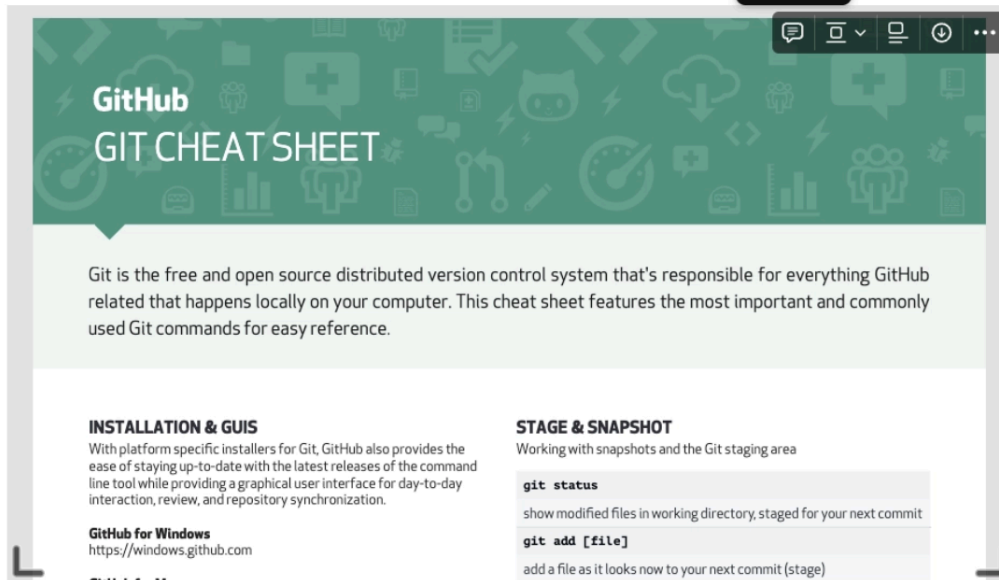


All of the files we created and worked on for this first sprint have been view files, as most of the functionality implemented has been navigation from one screen to another, a function that falls squarely under views in the MVVM architecture. Furthermore, many of these screens are simply laying

groundwork for later sprints (see the placeholder screens), during which one may assume supporting model and viewmodel files will be added.

## Agile Development





For spring 0.5 and 1.0, the team utilized Notion as our management tool to keep track of the progress of certain key tasks. We created categories such as Spring backing, in progress, and done to track the development and management of various project elements. This has helped us maintain a clear overview of our tasks and to-do's for each member as we progress to our deadlines. Furthermore, we utilized Notion's calendar feature to keep us aligned with deadlines from both Sprint 0.5 and 1 to ensure that the team stays on track as the days go by. In addition, we used Nation's ability to display files to upload a git cheat sheet onto our page. This was useful for our team as it allowed easy access to review any git commands and procedures to ensure smooth progress.

For Sprint 1, our team conducted regular scrum meetings both in person and online to maintain communication and progress. We met every Wednesday at 12:00 p.m., just before class, to discuss the status of our work, address any obstacles, and plan the tasks for the upcoming week. These meetings allowed us to locate which parts of the project required attention, making sure that everyone had a clear understanding of the priorities. We used this time to break down large tasks into smaller pieces and evenly distributed the workload among team members. This helped us maintain focus and stay aligned with the project's objectives. In addition to our in-person scrums, we held weekly meetings on Discord to stay connected throughout the week. These virtual check-ins were used for tracking progress, fixing any issues, and making adjustments to our projects.

## Implementation

[https://github.com/titanium-sodium/CS2340C\\_Team4/tree/main](https://github.com/titanium-sodium/CS2340C_Team4/tree/main)