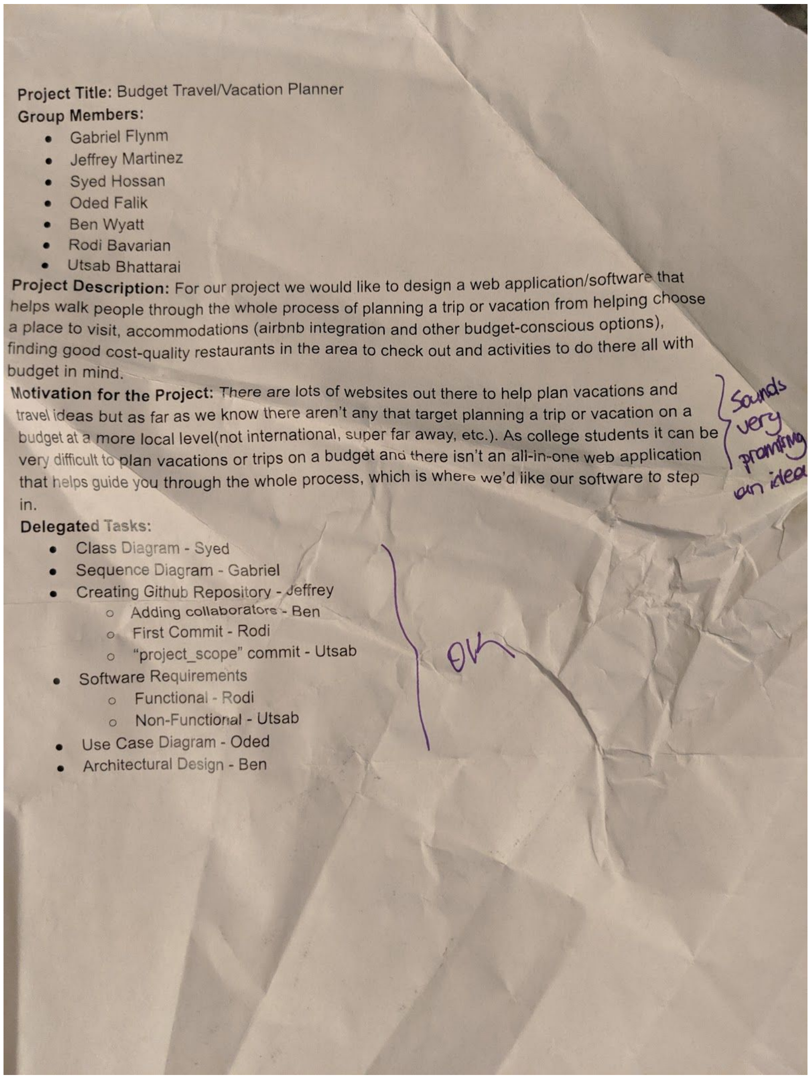
CS3354 Software Engineering
Final Project Deliverable 1

Budget Travel Planner

Gabriel Flynn, Jeffrey Martinez, Syed Hossan,
Oded Falik, Ben Wyatt, Rodi Bavarian, Utsab Bhattarai

1. Final Project Draft Description



Project Title: Budget Travel/Vacation Planner

Group Members:

- Gabriel Flynm
- Jeffrey Martinez
- Syed Hossan
- Oded Falik
- Ben Wyatt
- Rodi Bavarian
- Utsab Bhattarai

Project Description: For our project we would like to design a web application/software that helps walk people through the whole process of planning a trip or vacation from helping choose a place to visit, accommodations (airbnb integration and other budget-conscious options), finding good cost-quality restaurants in the area to check out and activities to do there all with budget in mind.

Motivation for the Project: There are lots of websites out there to help plan vacations and travel ideas but as far as we know there aren't any that target planning a trip or vacation on a budget at a more local level(not international, super far away, etc.). As college students it can be very difficult to plan vacations or trips on a budget and there isn't an all-in-one web application that helps guide you through the whole process, which is where we'd like our software to step in.

Sounds very promising an idea

Delegated Tasks:

- Class Diagram - Syed
- Sequence Diagram - Gabriel
- Creating Github Repository - Jeffrey
  - Adding collaborators - Ben
  - First Commit - Rodi
  - "project_scope" commit - Utsab
- Software Requirements
  - Functional - Rodi
  - Non-Functional - Utsab
- Use Case Diagram - Oded
- Architectural Design - Ben

Ok

Text transcription of the draft:

Project Description: For our project we would like to design a web application/software that helps walk people through the whole process of planning a trip or vacation from helping choose a place to visit, accommodations (airbnb integration and other budget-conscious options), finding good cost-quality restaurants in the area to check out and activities to do there all with budget in mind.

Motivation for the Project: There are lots of websites out there to help plan vacations and travel ideas but as far as we know there aren't any that target planning a trip or vacation on a budget at a more local level(not international, super far away, etc.). As college students it can be very difficult to plan vacations or trips on a budget and there isn't an all-in-one web application that helps guide you through the whole process, which is where we'd like our software to step in.

Feedback Addressal: Continue with the project as described in our draft, no changes/requests were suggested.

1. **Setting up a Github repository:**

   Github repository link: https://github.com/jeffthemaestro/3354-BudgetTravel

2. **Delegation of tasks**
   - Class Diagram - Syed
   - Sequence Diagram - Gabriel
   - Creating Github Repository - Jeffrey
     - Adding collaborators - Ben
     - First Commit - Rodi
     - “project_scope” commit - Utsab
   - Software Requirements
     - Functional - Rodi
     - Non-Functional - Utsab
   - Use Case Diagram - Oded
   - Architectural Design - Ben

3. **Which software process is employed in the project and why?**

   We are going to employ the incremental process model, specifically Agile methodology/development, as it’s very important to quickly figure out what works or doesn’t work as we add each feature to our web application. Incrementally adding the features allows us to get feedback from stakeholders in order to improve it early on instead of releasing it and then figuring out the way we implemented our idea wasn’t effective or meet the wants of our stakeholders.

4. **Software requirements including**

   a. **Functional requirements**

      i. Find all available tickets for destination

      ii. Compare ticket prices

      iii. Find travel routes which accommodate for potential midway stops

      iv. List places to eat, places to stay, entertainment at destination

      v. Display travel plans in different ways(i.e. List format, Visualize using a map)

   b. **Non-functional requirement**s

      1. Users are asked to follow appropriate password specifications and verify their email while creating a new account.

      2. The software is portable with easy migration to other platforms.

      3. The software will not disclose any personal information about customers.

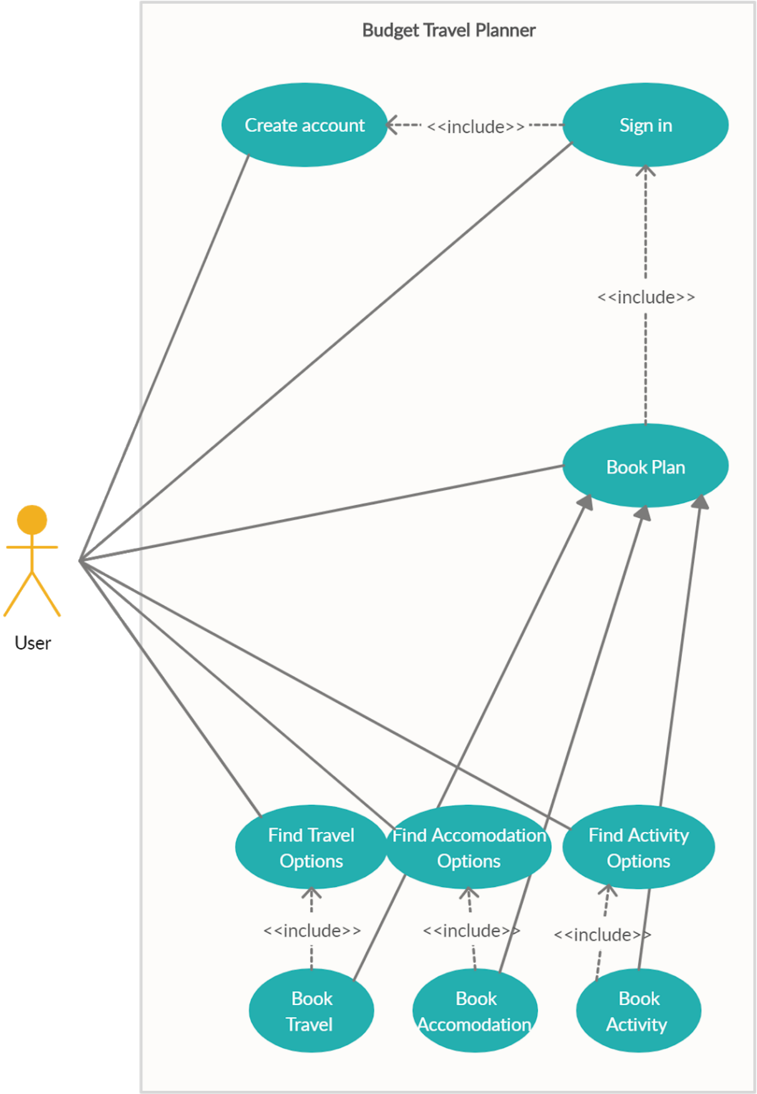
      4. The usability is simple and straightforward since the app is designed for all age range as well as non-technical users.

5. The performance is instantaneous with less than two seconds of response time with an exception of integration of Airbnb which will depend on request and response time of their API.

6. The app is made available for free in google play store and the system will be available to use 99% of the time during a month.

7. The app is easily maintainable as any bug fixes or errors will be fixed with a simple downloadable update.

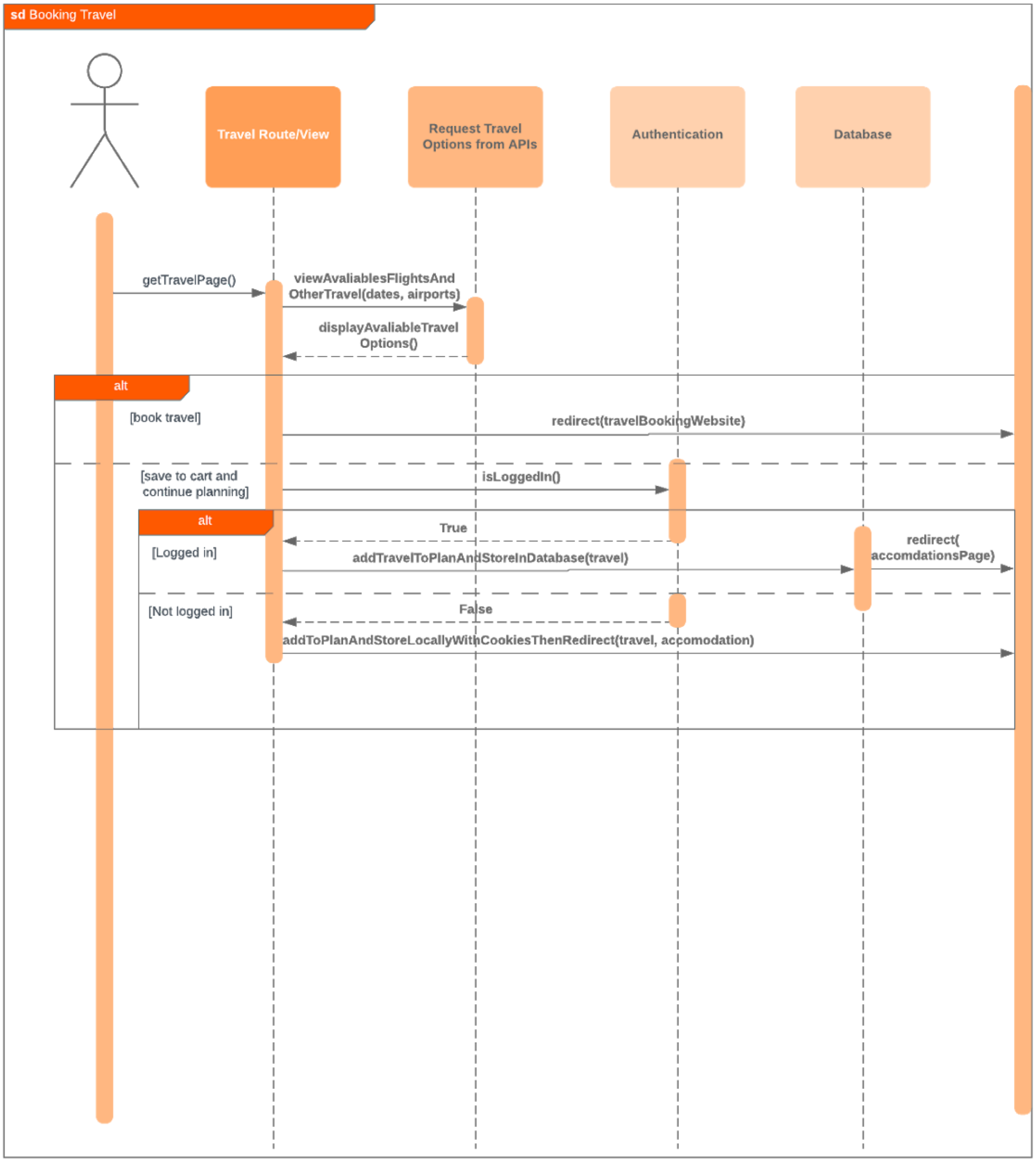
5. Use case diagram



6. Sequence diagrams

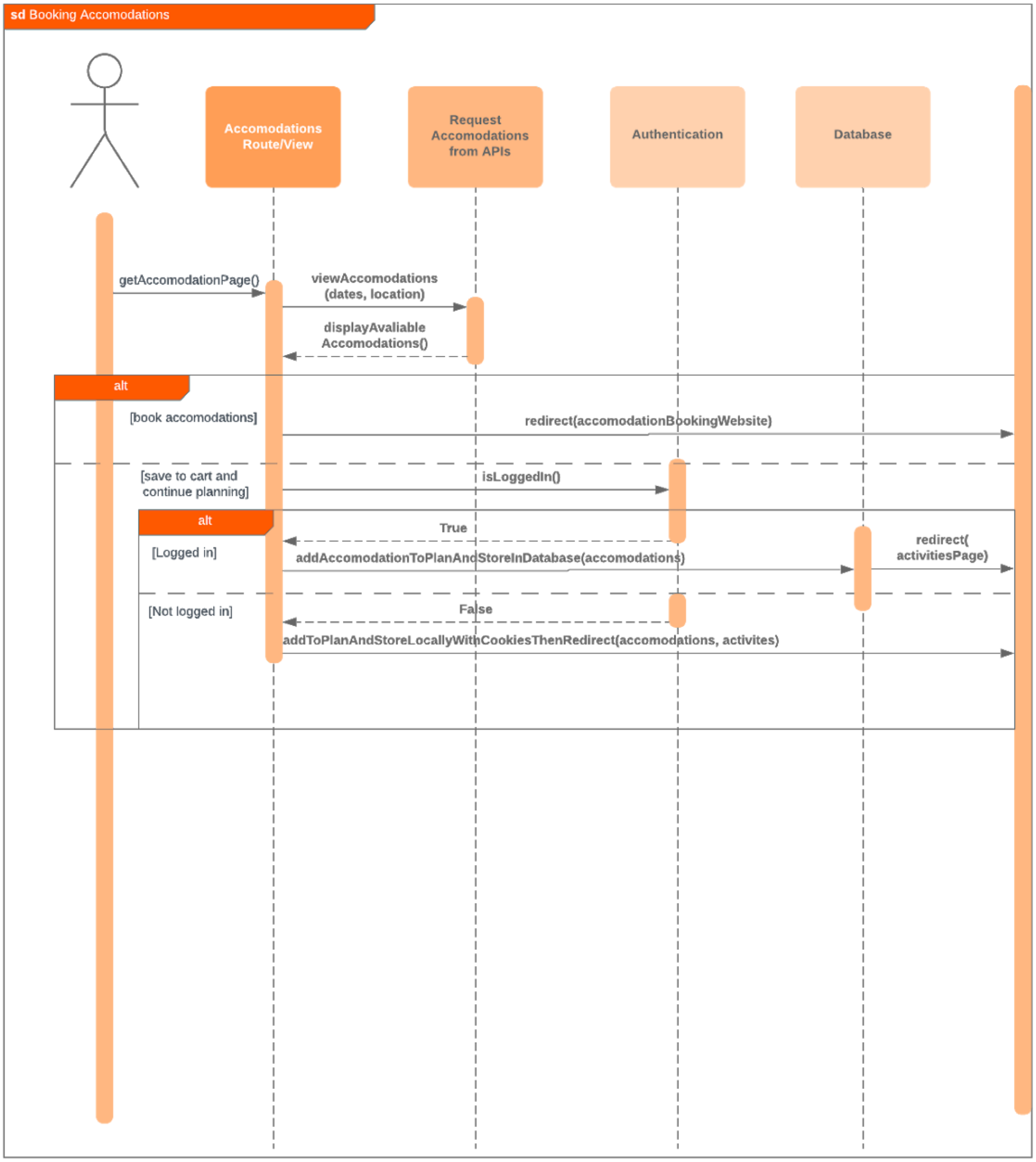
Travel Page Sequence Diagram

Gabriel Flynn | March 13, 2020


sd Booking Travel
Travel Route/View
Request Travel Options from APIs
Authentication
Database
getTravelPage()
viewAvaliablesFlightsAnd OtherTravel(dates, airports)
displayAvaliableTravel Options()
alt
[book travel]
redirect(travelBookingWebsite)
[save to cart and continue planning]
isLoggedIn()
alt
True
[Logged in]
addTravelToPlanAndStoreInDatabase(travel)
redirect(accomdationsPage)
[Not logged in]
False
addToPlanAndStoreLocallyWithCookiesThenRedirect(travel, accomodation)

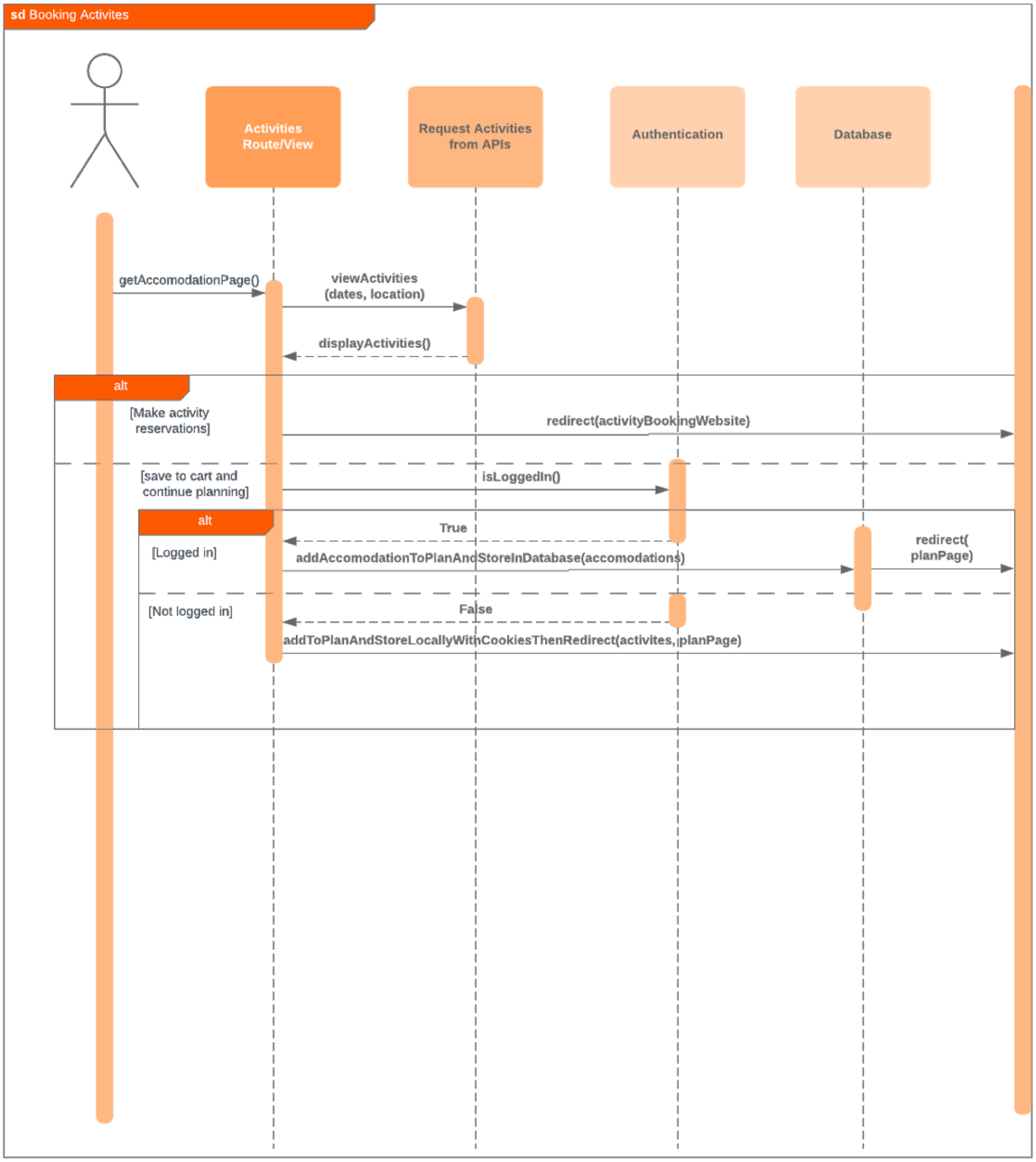
Accomodations Sequence Diagram

Gabriel Flynn | March 13, 2020


sd Booking Accomodations
Accomodations Route/View
Request Accomodations from APIs
Authentication
Database
getAccomodationPage()
viewAccomodations (dates, location)
displayAvaliable Accomodations()
alt
[book accomodations]
redirect(accomodationBookingWebsite)
[save to cart and continue planning]
isLoggedIn()
alt
True
[Logged in]
addAccomodationToPlanAndStoreInDatabase(accomodations)
redirect(activitiesPage)
[Not logged in]
False
addToPlanAndStoreLocallyWithCookiesThenRedirect(accomodations, activites)

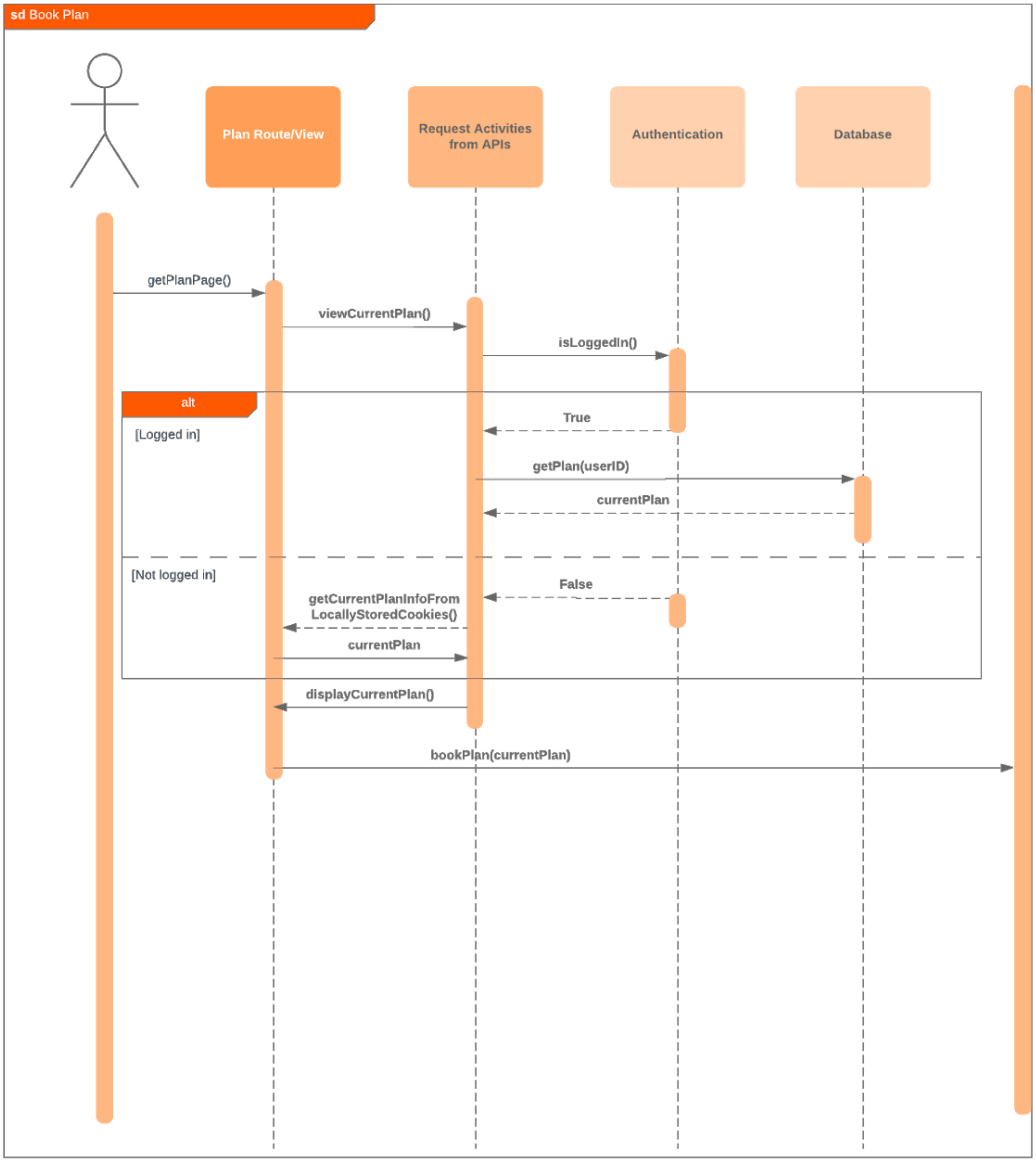
Activities Sequence Diagram

Gabriel Flynn | March 13, 2020


sd Booking Activites
Activities Route/View
Request Activities from APIs
Authentication
Database
getAccomodationPage()
viewActivities (dates, location)
displayActivities()
alt
[Make activity reservations]
redirect(activityBookingWebsite)
[save to cart and continue planning]
isLoggedIn()
alt
True
[Logged in]
addAccomodationToPlanAndStoreInDatabase(accomodations)
redirect(planPage)
[Not logged in]
False
addToPlanAndStoreLocallyWithCookiesThenRedirect(activites, planPage)

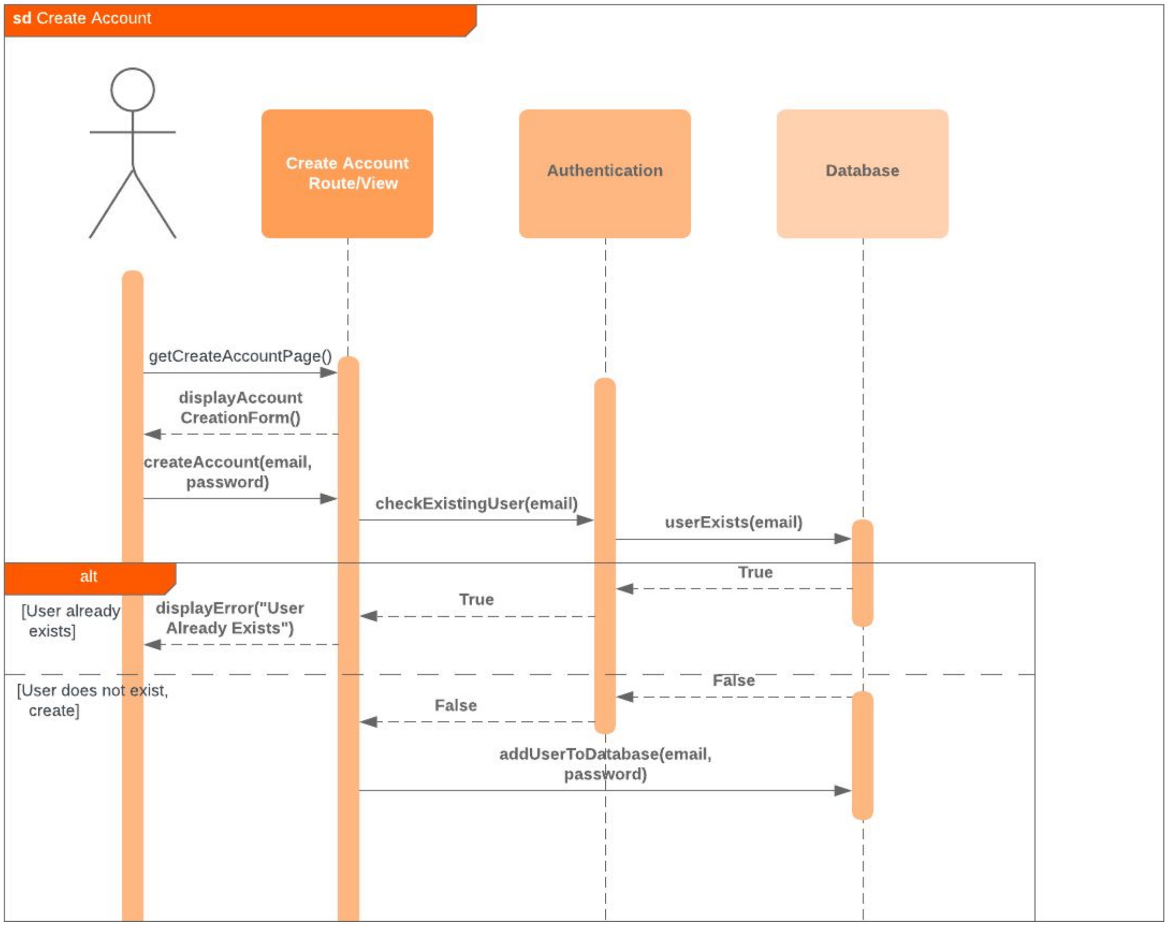
Plan Sequence Diagram

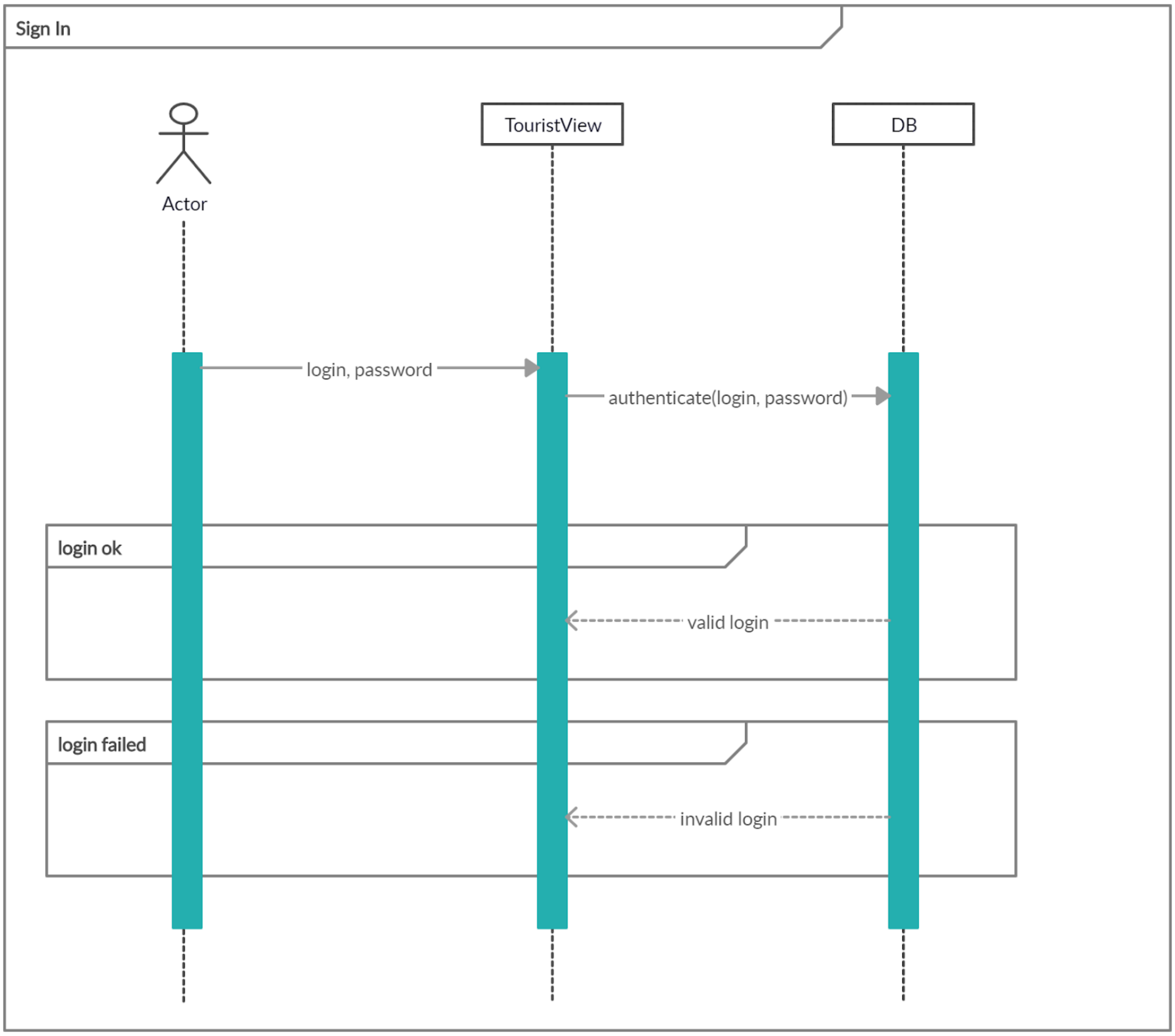
Gabriel Flynn | March 13, 2020



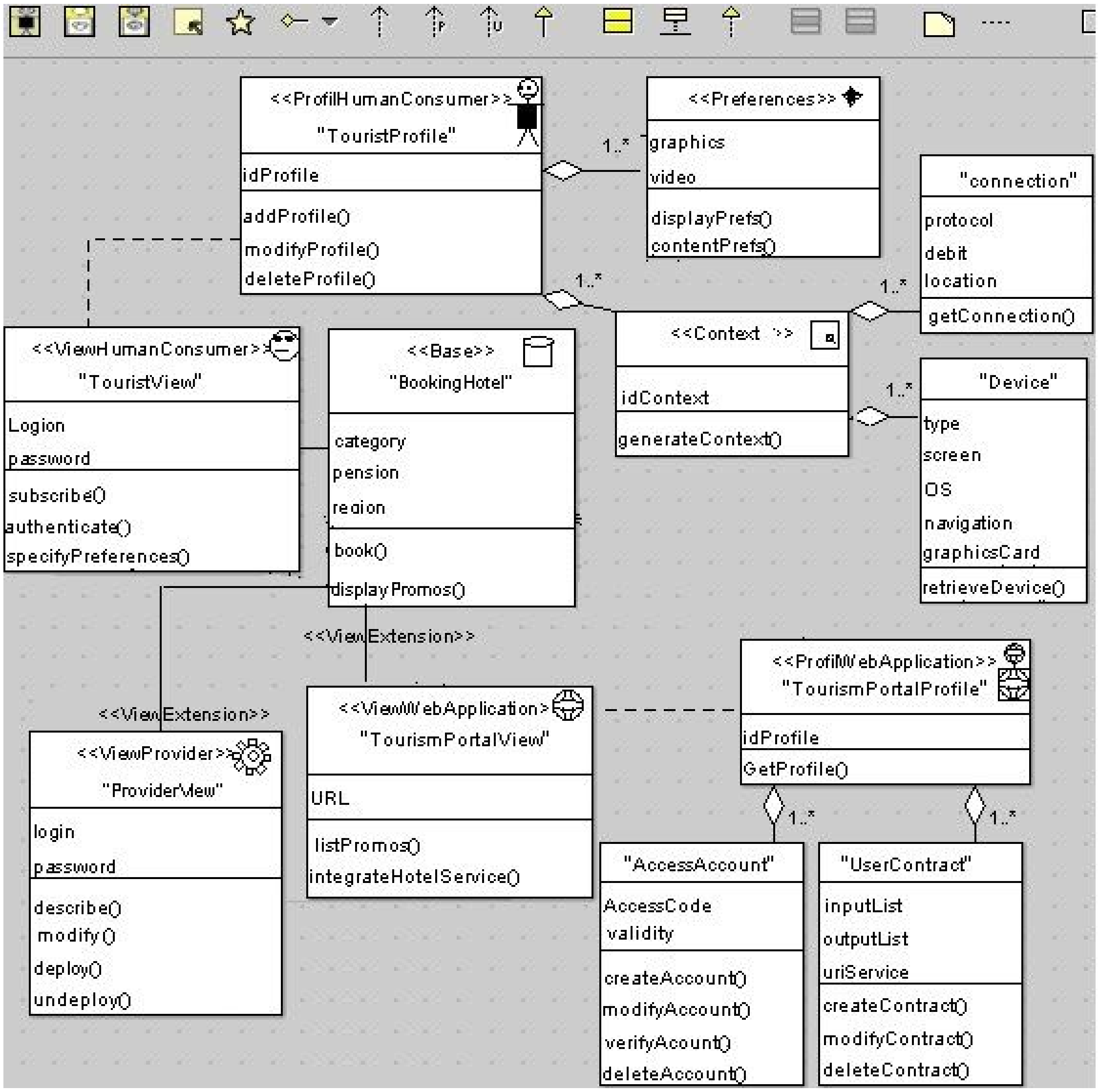
Create Account Sequence Diagram

Gabriel Flynn | March 13, 2020

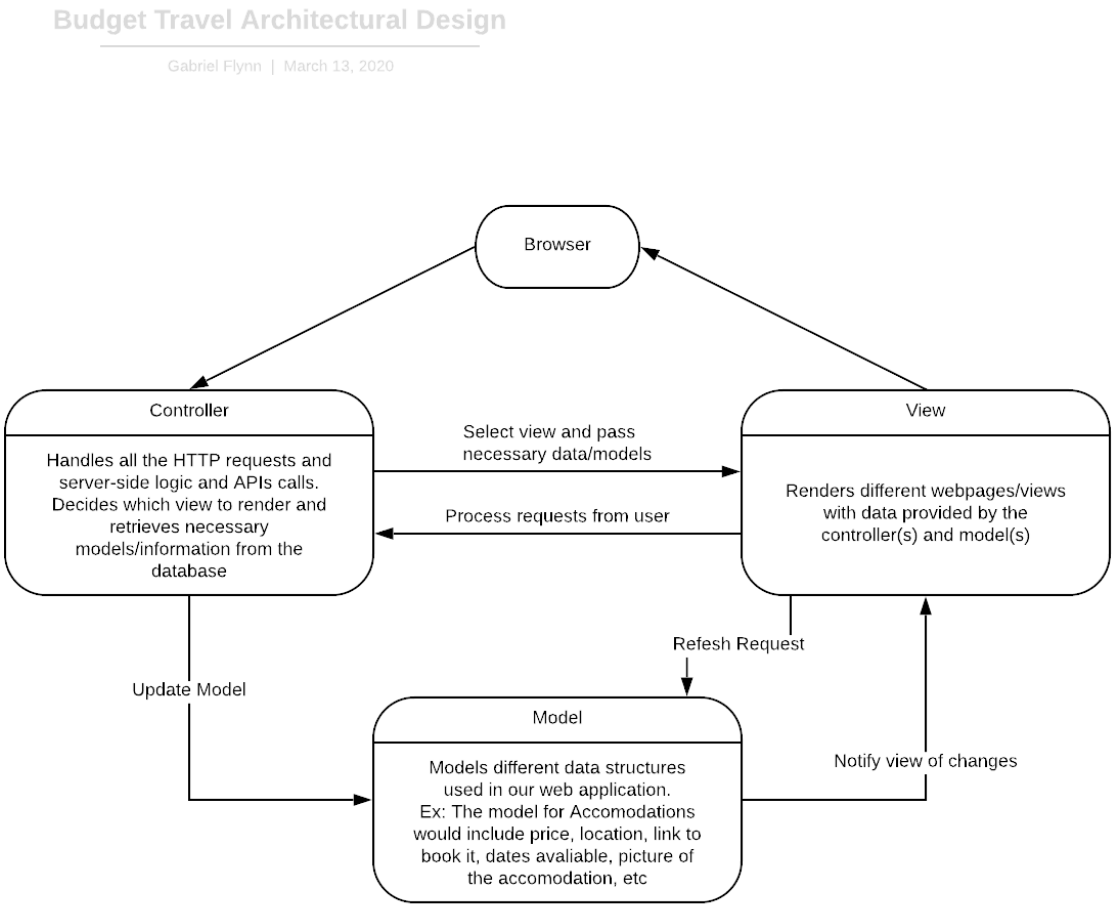

sd Create Account
Create Account Route/View
Authentication
Database
getCreateAccountPage()
displayAccount CreationForm()
createAccount(email, password)
checkExistingUser(email)
userExists(email)
alt
True
True
[User already exists]
displayError("User Already Exists")
False
[User does not exist, create]
False
addUserToDatabase(email, password)


Sign In
Actor
TouristView
DB
login, password
authenticate(login, password)
login ok
valid login
login failed
invalid login

7. **Class diagram**



8. **Architectural design**



Budget Travel Architectural Design

Gabriel Flynn | March 13, 2020

Browser
Controller
Handles all the HTTP requests and server-side logic and APIs calls. Decides which view to render and retrieves necessary models/information from the database
Select view and pass necessary data/models
Process requests from user
View
Renders different webpages/views with data provided by the controller(s) and model(s)
Refesh Request
Update Model
Model
Models different data structures used in our web application. Ex: The model for Accomodations would include price, location, link to book it, dates avaliable, picture of the accomodation, etc
Notify view of changes

We decided on the Model-View-Controller architecture for our budget travel web application as it would make the most sense and it supports more features that would make development easier and more efficient. There are plenty of MVC frameworks available for use in tons of different programming languages. We also have a lot of different data being stored, from the transport(travel), to accommodations, to activities and more. Having the ability to easily separate the data that is stored in each component and the view that the user sees is very important. As this software improves and tries to integrate more and more different options and APIs to help people plan and book their budget travel, it’s important for it to be easy to update these different components, which is where the MVC architecture really shines. Changing the views that are rendered, adding more interactions between components and users, and adding more data to our models is very important in a web application that is constantly improving and very important for the Software Development Lifecycle.