Name of Project: Book a Cruise
Website Name: Cruising
Group Number: 1
Team Members: Randall Byrd, Riyazh Dholakia, Syed Kalam, Sujal Patel, Biprojoyti Paul
Semester: Spring 2019
Date of Submission: February 17th, 2019

Planning and Schedule:

Work Breakdown Structure

Assignee Name	Email	Task	Duration (hours)	Dependency	Due date	Note
Riyazh Dholakia (**Coordinator**)	rdholakia1@student.gsu.edu	Use Cases, Requirements, Problem Statement, Format Project, Database Specification and Analysis	4 hours	Everyone's work to format project, Microsoft Visio Download, Use Case(s)	2/17/19	100%
Sujal Patel	spatel287@student.gsu.edu	Use Cases, Requirements, GitHub Board, Class Diagram	4 hours	Microsoft Visio Download, Use Case(s)	2/17/19	100%
Randall Byrd	rbyrd11@student.gsu.edu	Use Cases, Use Case Diagram, Requirements, Database Specification and Analysis	4 hours	Use Case(s), Microsoft Visio Download, Use Case(s)	2/17/19	100%
Syed Kalam	skalam1@student.gsu.edu	Use Cases, Use Case Diagram, Requirements, Class Diagram, System Modeling	4 hours	Use Case(s), Microsoft Visio Download, Use Case(s)	2/17/19	100%
Biprojoyti Paul	bpaul5@student.gsu.edu	Use Cases, Problem Statement, Requirements, Class Diagram	4 hours	Microsoft Visio Download, Use Case(s)	2/17/19	100%.

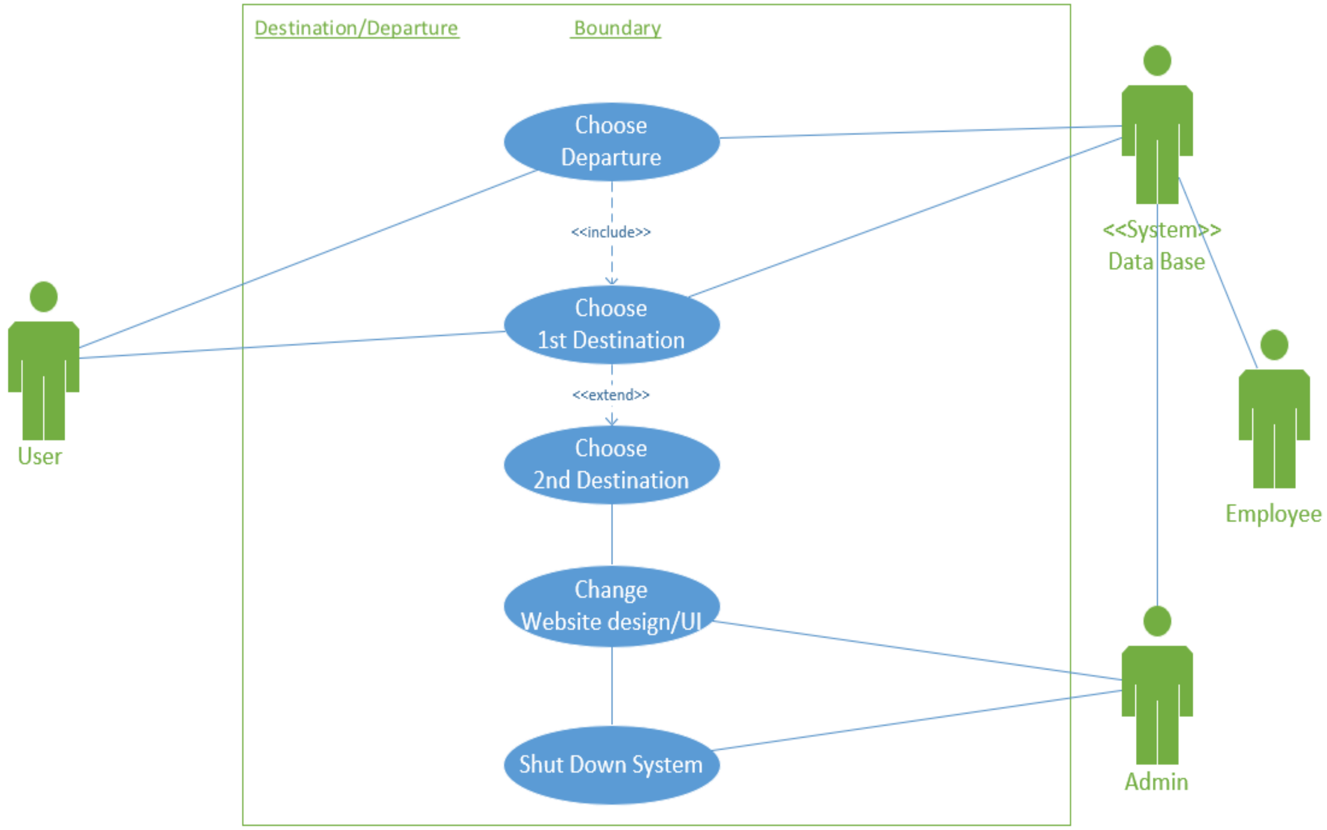
Problem Statement:

People who want to go on vacation, have fun, new experience, explore the world, love being on a boat, traveling, food, etc. There are many reasons to go on a cruise. It is also better price wise, compared to flying. Some people feel safer on water rather than flying to places. The audience is endless, since anyone can access site. Our competitors are doing this now. The audience can also be controlled by the market and the client. Sometimes, it is hard to find a cruise booking place. Nowadays, many people want to go to their desired place and wanted to be close to the water and be with the family. The valuable thing is that the cruise gives them all things to a reasonable price. Cruise trips the best trips ever, that incorporate boats, attractions, clubs, family games, and more. People can book plane, car, or go by train, bus, but the fact that Cruise is different than others, tells you more people are will experience this. Anybody would love to be part of an amazing scenery and experience. Other alternatives can also be to book from competitors' site. The top-level objective is building a product to officiate, booking a cruise. Differentiators from other products is out will be simple, better UX, and other various packages other competitors do not offer. There is a lot of websites and apps already been made on this project. If somebody Googles, they will find many different places to book cruises. We have seen a lot of websites where the look of the website does not appeal as much so that the customer will not be impressed to book the cruise from their website. We must make a front-end and back-end/database which will connect to each other. We think we have enough resources to make that possible. If we need help, we can use Google, YouTube, and Udemy courses to sharpen our skills. The front-end system we can built by HTML, CSS, and JavaScript. To make it responsive we will use some jQuery. For database, we will use Firebase. On backend we will use node, JavaScript, hopefully Firebase will take care of server.

It will be fun and challenging to make such kind of project. The fact that we can build a system, which can book ticket with given customer information will be mined blowing, and we can use it later any similar system if we want. This technical idea is common process, hopefully we can introduce new tech into the scene. The fact that we can build a system, which can book ticket with given customer information will be mined blowing, and we can use it later any similar system if we want. This technical idea is common process, hopefully we can introduce new tech into the scene. This is also help us build our skills and gives a good resume project. These skills help us in our future career.

Requirements:

Use Case Diagram for Use Case #3:



Use Cases:

Use Case #: 1

Name of the Use Case: User Sign Up

Actors: User, Admin, and Database

Description:

- The user will click on sign up option on login page.
- After clicking sign up, the user will have to fill a form out, with inputs in requirements.
- The form boxes are “includes” so they are all required to be filled.
- After the inputs are complete, the user will have to hit submit.
- If successful, the user will get success message.
- If unsuccessful, the user will a try again message to try again.
- The information will save in database if successful.
- After successful they will go to login in page to login in with their email and password.
- Admin can change the look and feel of website and anything in database.

Exception Path: If the user already has an account with us, they will be prompted to Login.

Alternate Path: There is only one path in this case, unless error to try again.

Pre-condition: Browser, Mouse, Keyboard, Internet Connection.

Post-condition: Login in page after signing up is complete.

Use Case #: 2
Name of the Use Case: User Login
Actors: User, Admin, and Database
Description:

- The user will be given the login page, once accessing out site.
- The user will enter email and password(includes).
- Recommended password option is an excludes option.
- After the inputs are complete, the user will have to hit submit.
- If successful, the user will end up on the destination/departure page.
- If unsuccessful, the user will a try again message to try again.
- The information will be validated with database.
- Admin can change the look and feel of website and anything in database like there email and password.

Exception Path: If the user does not have an account with us they will be prompted to Sign Up.
Alternate Path: There is only one path in this case (Departure/Destination site), unless error to try again.
Pre-condition: Valid account must be already made.
Post-condition: Entering our Destination/Departure page through browser after pre-condition has been met.

Use Case #: 3
Name of the Use Case: User selects Destination(s) and Departure
Actors: User, Employee, Admin, and Database
Description:

- A user should be able to select a Destination(s) and Departure selection for their cruise.
- A Destination must include a Departure.
- A Destination can exclude multiple Destinations.
- Employee and Admin should be able to change the Destination and Departure locations through the database.
- Admin can turn off page and change the look and feel of it.

Exception Path: If the user already had a Destination and Departure selected and we saved it in a cookie/session.

Alternate Path: There is two alternate paths in this case. The one path is the “normal” path of the user to select Destinations and Departure. The other path is Employee and Admin, being able to change Destination and Departure in database. Admin being able to turn off system and page.
Pre-condition: Browser, mouse, keyboard, internet connection, and valid login account.
Post-condition: Next page which is to select and room.

Use Case #: 4

Name of the Use Case: User selects Room

Actors: User, Database, and Employee

Description:

- The user will have to click the room options tab.
- Upon selection, the user will have to specify the number of adults and or children that will be in attendance
- After, the user will be prompted to select from a list of three-room types: Family room, Executive suite, and Presidential suite
- After giving the desired room type, the database will look to see if that room is available, and if not it will list of other room options that are.
- If there is no vacancy for the room, then an error message may be thrown, and a list of other available room options may be prompted on the screen, else finding the desired room will generate a success.
- Each room will have a price to determine of the cruise.
- Employee will change room types and room price in the Database system.

Exception Path: If the user already had a Room selected and we saved it in a cookie/session.

Alternate Path: The alternative path will be the room type options (presidential suite, executive suite, and family room)

Pre-condition: Desired room type, being logged in, and after selecting Destination/Departure.

Post-condition: Room selection is complete and then you go to the next page which is to choose Dates.

Use Case #: 5

Name of the Use Case: User selects Dates

Actors: User, Admin, and Database

Description:

- The user will select the dates option
- After clicking, the user will select the start and end dates of his or her trip
- The database will then look up respective cruises during that time
- if for any reason a cruise is not available then the database will generate an error message to user. if there is availability then a list of cruises will be shown

Exception Path: If the user already had Dates selected and we saved it in a cookie/session.

Alternate Path: There is only one path, unless the dates yield error then the user will be prompted to enter new dates

Pre-condition: knowing the dates, desired room type, being logged in, and after selecting Destination/Departure

Post-condition: the dates are chosen, and the user will be sent to the passenger information screen.

Use Case #: 6

Name of the Use Case: User enters Passenger(s) Information

Actors: User and Database

Description:

- The user will select the Passenger(s) Information option
- The database will allocate information box(s) for each passenger in attendance, the number of passengers is retrieved from the select a room's page, given by the user.
- After, the user will be prompted to give the first name, last name and date of birth of each passenger starting from passenger 1
- If any passenger box is left unfilled then an error message to complete the form will be shown.
- if everything is complete the user will click next and will be brought to the payments page.

Exception Path: If the user already had a Passenger info entered before, we will have is saved in our Database.

Alternate Path: there is only one path on this page and if there is error the user will have to try again

Pre-condition: having passenger's information, knowing the dates, Desired room type, being logged in, and after selecting Destination/Departure

Post-condition: the information is given, and the user is sent to the payments page.

Use Case #: 7

Name of the Use Case: User Pays for the Cruise

Actors: User and Database

Description:

- The user will enter the payment page.
- User must put all his information like first name, last name, address, country, city state, region, zip code, telephone number, email address.
- Then user must put the credit/debit card number.
- Our system will be able to take the American Express card, Discover, Master Card, Visa.
- User should be able to provide the security code, expiration date, Billing Address.
- The information will be sent to our Database system for the payment to be processed and saved for next time.

Exception Path: If they have a saved payment method already on our Database system.

Alternate Path: The user can call instead to book the cruise.

Pre-condition: User should confirm all other pages before coming here, user should know the dates, user must have money in the card.

Post-condition: If all the required fields of payment page is done then user will go to confirmation page.

Use Case #: 8

Name of the Use Case: User gets an Order Confirmation

Actors: User and Database

Description:

- After the payment page, the user would see a confirmation page.
- The confirmation page will have a confirmation number.
- User will have a confirmation email sent out to the given email address.

Exception Path: If the confirmation page is closed they can access it in their email.
Alternate Path: There is no alternate path for this case. The user should be able to see the confirmation page after the payment.
Pre-condition: The user must pay through his credit/debit card.
Post-condition: User should must get a confirmation number and a email confirmation.

Use Case #: 9

Name of the Use Case: User can Logout

Actors: User

Description:

- After the order confirmation page, the user will be prompted to logout.
- If the user will have the option to go back to Departure/Destination page to book another cruise.

Exception Path: The other path can be if the user automatically gets logged off when the session/cookie to done.
Alternate Path: For user either they can log out or keep going through our website.
Pre-condition: Browser, Mouse, Keyboard, Internet Connection.
Post-condition: Logging out page after reservation is complete.

System Requirements:

Requirement #: 1

Use Case #: 1

Name: Sign Up

Introduction: Easy look and feel for the form for the user to sign up with so they can login to our website, to book a cruise.
Rationale: Sign up process for user to login into the site, and to start booking a cruise.
Input: First Name, Last Name, Email, Password, Confirm Password, Phone Number, Full Address, DOB
Requirement Description: The user should be able to sign up. This is only possible with all requirements are met.

1. All fields must be completed.
2. Requirements from the system are met too.

Output: A successful screen, then routed to login to log back in.
Test Cases: TBD

Requirement #: 2

Use Case #: 2

Name: Login

Introduction: Easy look and feel for the 2-input form for the user to login to go to our Destination/Departure page.

Rationale: Login process for user to login into the site, and to start booking a cruise.
Input: Email and password
Requirement Description: The user should be able to sign up. This is only possible with all requirements are met.

1. All fields must be completed.
2. Requirements from the system are met too.
3. Database validation must be met also.

Output: Routed to our Destination/Departure page.
Test Cases: TBD

Requirement #: 3

Use Case #: 3

Name: Destination(s) and Departure

Introduction: Destination and Departure page to select where the user wants to go for their cruise. The page will be simple and easy to choose from.
Rationale: Destination/Departure page, this is the first step in the cruise booking process after login validation.
Requirement Description: Destination/Departure page with easy check boxes.
There should at least 1 Destination selected

1. There should be only 1 Departure selected.
2. All information is pulled from Database when should locations.
3. The user can select places that are not offered. They should be unselecting (grayed out). checking with Database constantly.

Output: Routed to Room page after all correct selections have been made.
Test Cases: TBD

Requirement #: 4

Use Case #: 4

Name: Room

Introduction: Room selection is made readily simple for customers to choose desired room options.
Rationale:
Input: Number of people in attendance and desired room type
Requirement Description:

1. User must input a valid number of people for correct search results for example 100 guest for one room would be far too many.
2. Database finds appropriate rooms requested by customers

Output: A room selection success is affirmed, and the user will be routed to the dates page.
Test Cases: TBD

Requirement #: 5

Use Case #: 5

Name: Dates

Introduction: choosing respective dates will be easy and user friendly
Rationale:
Input: Start date and end date
Requirement Description:

1. Both start and end dates must be given to generate a search result
2. the database outputs an error or a list of cruises.

Output: A successful screen, then routed to the passenger information page.
Test Cases: TBD

Requirement #: 6

Use Case #: 6

Name: Passenger Information

Introduction: Entering passenger(s) information will be made easy and efficient
Rationale: the database will have the exact amount of necessary spaces for each passenger in attendance, so the user does not have to do it.
Input: first name, last name, and date of birth
Requirement Description:

1. All fields must be completed
2. Database validation must be met also.

Output: No errors will persist, and the user will then move on to the payments page
Test Cases: TBD

Requirement #: 7

Use Case #: 7

Name: Payments

Introduction: making a payment will be divergent for all users
Rationale: the system will support multiple banking platforms (Visa, Mastercard, etc.)
Input: the first and last name on the card, the credit or debit card number, the security code, expiration date, and billing address
Requirement Description:

1. All fields must be entered correctly.
2. Database must ensure security.

Output: The credit/debit card will be accepted, sending the user to the order confirmation page.
Test Cases: TBD

Requirement #: 8

Use Case #: 8

Name: Order Confirmation

Introduction: Order history will be safe and secure in the database
Rationale: The database will generate a unique confirmation code to respective users
Input: None
Requirement Description:

1. The Database system must save the order in the system.
2. The Database system must populate an order confirmation number and be then shown to the user on the screen.
3. The Database system will also have to email the confirmation number to the user.

Output: The user will receive an order confirmation number and will be prompted to Logout or got to Departure and Destination page.
Test Cases: TBD

Requirement #: 9

Use Case #: 9

Name: Logout

Introduction: Logging out will be click of button easy for the user after they reserved the cruise.
Rationale:
Input: Click of button
Requirement Description: User should Logout after done with our site.

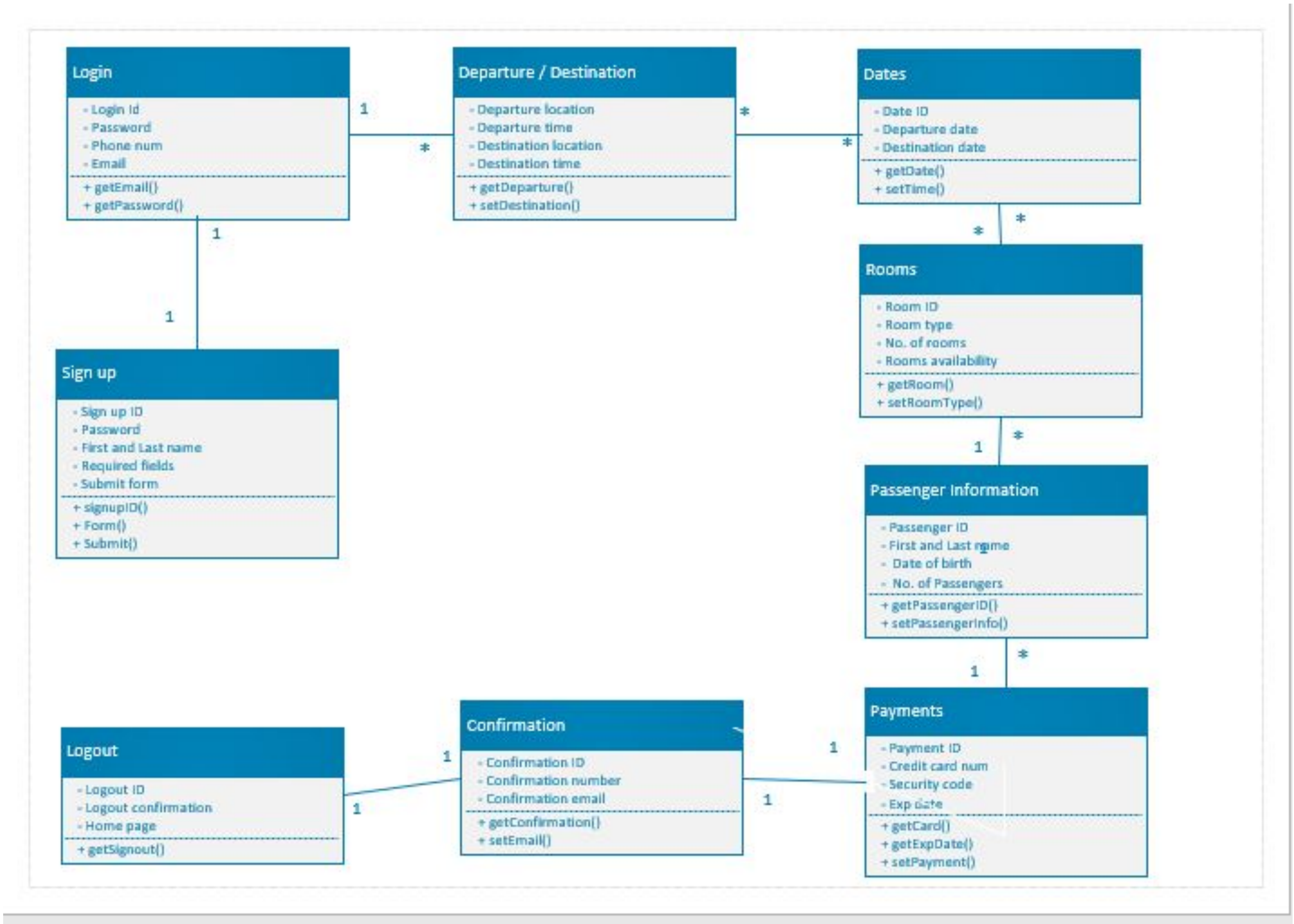
1. Give the user and option to logout.
2. If cookie/session is over the user should logout automatically.

Output: Logged out of the page, back to the login page.
Test Cases: TBD

System Modeling:

The objects are: sign up, login, departure/destination, dates, rooms, passenger information, payment, confirmation, and logout. The only association is that Payments have a confirmation. Sign up to login is one to one. Login to departure/destination is one to many. Departure/destination to dates is many to many. Dates to rooms is many to many. Rooms to passenger information is many to one. Passenger information to payments is many to one. Payments to confirmation is one to one. Confirmation to logout is one to one. The attributes of sign up are signup id, password, first and last name, required fields, and submit form. The attributes for login is login id, password, phone number, and email. The attributes for destination/departure are departure location, departure time, destination location, and destination time. The attributes for dates are date id, departure date, destination date. The attributes for rooms are room id, room type, number of rooms, rooms availability. The attributes of passenger information are passenger id, first and last name, date of birth, and number of passengers. The attributes of payments are payment id, credit card number, security code, and expiration date. The attributes for confirmation are confirmation id, confirmation number, and confirmation email. The attributes for logout are logout id, logout confirmation, and home page. The operations for signup are signupid(), form(), and submit(). The operations for login are getEmail() and getPassword(). The operations for departure destination are getDeparture() and setDestination. The operations for dates are getDate() and setTime(). The operations for rooms are getRoom() and setRoomtype(). The operations for passenger information are getPassengerId() and setPassengerinfo(). The operations for payment are getCard(), getExpirationdate(), and setPayment(). The operations for confirmation are getConfirmation and setEmail(). The operations for logout is getSignout().

Class Diagram:



Database specification and analysis:

Database management system: Firebase.

Parent Tables:

- Customers: customer_id (PK), customer_first_name, customer_last_name, customer_phone_number, customer_email_address, customer_password, customer_address, customer_city, customer_state, customer_zipcode, customer_dob

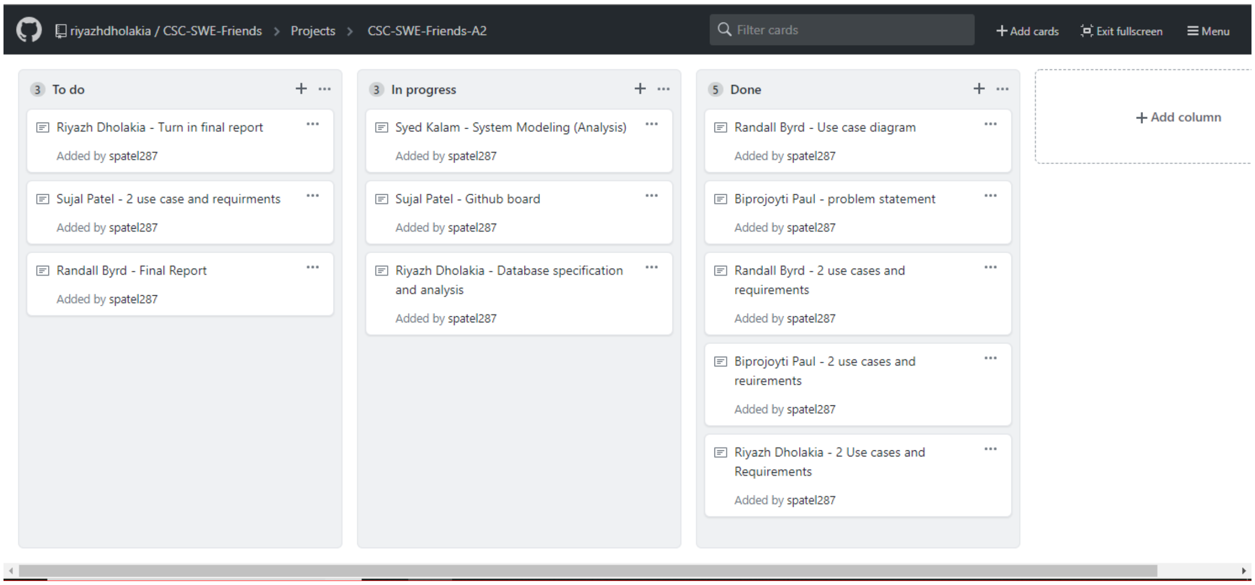
- Cruises: cruise_id (PK), cruise_destination, cruise_departure
- Dates: date_id (PK), cruise_date_start, cruise_date_end
- Passengers: passenger_id (PK), passenger_first_name, passenger_last_name, passenger_phone_number, passenger_email_address, passenger_dob
- Rooms: room_id (PK), room_type, room_price
- Payments: payment_id (PK), credit_card_num, credit_card_exp
- Order_confirmations: order_confirmation_id (PK), order_confirmation_number

Child Tables:

- Customers_cruises: customer_id (FK) and cruise_id (FK)
- Cruises_dates: cruise_id (FK) and date_id (FK)
- Cruises_roooms: cruise_id (FK) and room_id (FK)
- Payments_order_confirmation: payment_id (FK) and order_confirmation_id (FK)
- Customers_passengers: customer_id (FK) and passenger_id (FK)

Appendix:

GitHub Project Board:



3 To do

Riyazh Dholakia - Turn in final report
Added by spatel287

Sujal Patel - 2 use case and requirments
Added by spatel287

Randall Byrd - Final Report
Added by spatel287

3 In progress

Syed Kalam - System Modeling (Analysis)
Added by spatel287

Sujal Patel - Github board
Added by spatel287

Riyazh Dholakia - Database specification and analysis
Added by spatel287

5 Done

Randall Byrd - Use case diagram
Added by spatel287

Biprojoyti Paul - problem statement
Added by spatel287

Randall Byrd - 2 use cases and requirements
Added by spatel287

Biprojoyti Paul - 2 use cases and reuirements
Added by spatel287

Riyazh Dholakia - 2 Use cases and Requirements
Added by spatel287

GitHub Link: https://github.com/riyazhdholakia/CSC-SWE-Friends