SW Engineering CSC648/848

Application Name: DestiGo
Section 04, Team 06

Name	**Role**
Sahej Tuli	**Team Lead**
Ray Dela Cruz	**Front-End Lead**
Jessica Christine Rosero	**Back-End Lead**
Navjot Singh	**Git Master**
Faheemah Shaikh	**Scrum Master**
Ryan Tong	**Full-Stack Developer**

Milestone 2

October 11, 2023

Revision Table			
Revision	**Date**	**Author(s)**	**Description of Changes**
1	09/27/2023	Team Members	Initial Draft
2	10/11/2023	Team Members	Milestone 2

1. Data Definitions V2

Primary Data Name	**Definition**	**Usage**
Location	Where flights are departing and arriving, hotels, and events.	Used to sort data of other categories of data that are local to where the user is going. ● selectLocation ● changeLocation ● saveLocation
Users	The person who has a registered account on application.	Required to be able to use the application. ● user_id ● user_name ● password ● location ● use_avatar ● user_description
Flight	Flight information between locations including cost, and time.	Search by location and date.
Hotel	Hotel information including location, cost, and date.	Search by location and date.
Events	These would be points of interest around location. These recommendations might drive interest in planning an itinerary and include description, address, time, and cost.	Search by location.
Friends	This would be a list where users can group who they want to communicate with.	● addFriend ● removeFriend
Private Text	This is where correspondence with friends would happen. They can discuss their plans for itineraries.	● sendMessage ● recieveMessage ● name ● chat_ID ● photo_filepath ● post_filepath ● saveMessage
Community (Blog)	This is where users can openly communicate with	● blog_id ● blog_title

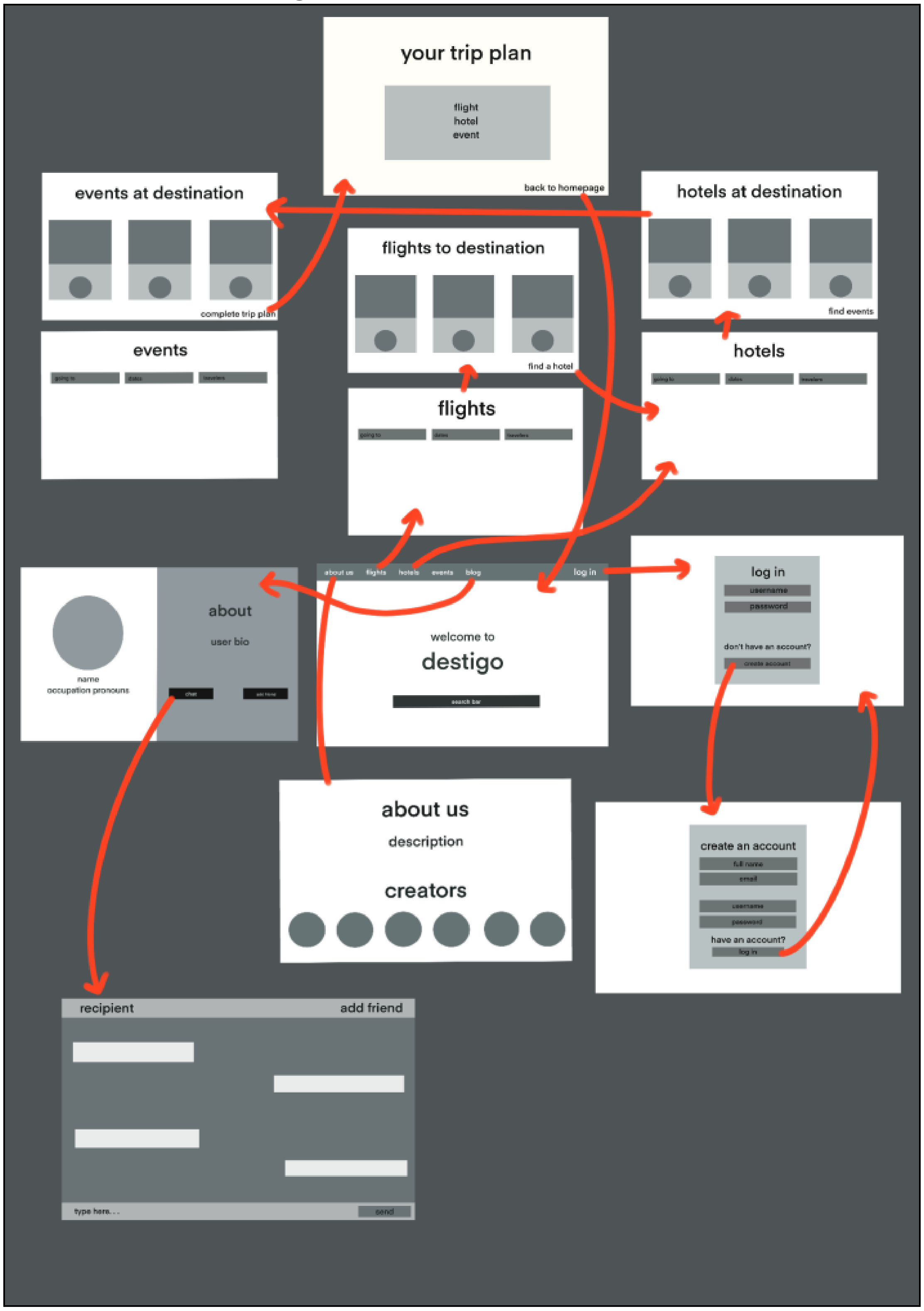
	all the other users of the application.	● blog_description ● blog_category ● post_date ● like_count ● view_count

2. Functional Requirements V2

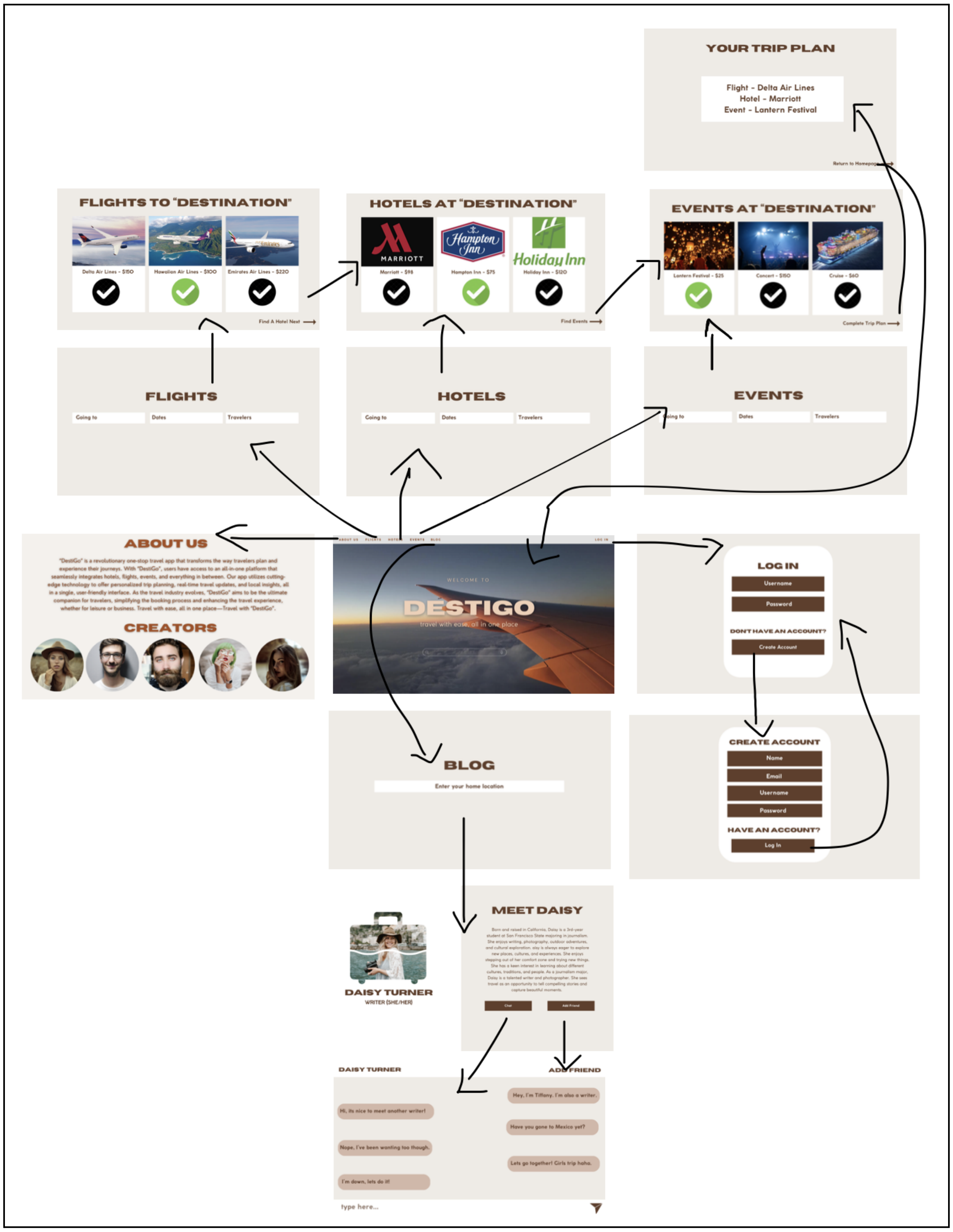
Functional Requirement Description	**Details**
Users can have their own personal account. **High Priority**	1. Users can register for a new account 2. Users can log in to their registered account
Users have access to flight information for planning trips. **High Priority**	1. Users can search by location and date.
Users have access to hotel information for lodging during trips. **High Priority**	1. Users can search by location and date
Users can see recommended points of interest at their trip location. **High Priority**	1. Users can search by location
Users can keep a list of friends and communicate with them via private messages. **High Priority**	1. Users can add/ remove friends 2. Users can private message friends. 3. users will be alerted to received messages
Users can communicate openly with all users of the application through the blog about any aspects or experiences of their trip. **Medium Priority**	1. Users can post. 2. Users can reply to posts 3. Users can view posts 4. Users can like posts

3. UI Mockups and UX Flows

Black and White Wire Diagram



GUI Prototype



"DestiGo" is a revolutionary one-stop travel app that transforms the way travelers plan and experience their journeys. With "DestiGo", users have access to an all-in-one platform that seamlessly integrates hotels, flights, events, and everything in between. Our app utilizes cutting-edge technology to offer personalized trip planning, real-time travel updates, and local insights, all in a single, user-friendly interface. As the travel industry evolves, "DestiGo" aims to be the ultimate companion for travelers, simplifying the booking process and enhancing the travel experience, whether for leisure or business. Travel with ease, all in one place—Travel with "DestiGo".

Born and raised in California, Daisy is a 3rd-year student at San Francisco State majoring in journalism. She enjoys writing, photography, outdoor adventures, and cultural exploration. aisy is always eager to explore new places, cultures, and experiences. She enjoys stepping out of her comfort zone and trying new things. She has a keen interest in learning about different cultures, traditions, and people. As a journalism major, Daisy is a talented writer and photographer. She sees travel as an opportunity to tell compelling stories and capture beautiful moments.

Hey, I'm Tiffany. I'm also a writer.

Hi, its nice to meet another writer!

Have you gone to Mexico yet?

Nope, I've been wanting too though.

Lets go together! Girls trip haha.

I'm down, lets do it!

4. High level Architecture, Database Organization

Users Table

Column	Data Type	Description
user_id (PK)	Integer	Primary Key for User
first_name	String	User's First Name
last_name	String	User's Last Name
username	String	User's Username
email	String	User's Email
password	String	User's Password (hashed)
picture	String (URL)	URL to User's Profile Picture

Friends Table

Column	Data Type	Description
friend_id (PK)	Integer	Primary Key for Friend Relationship
user_id (FK)	Integer	User's ID (Foreign Key referencing Users)
friend_user_id	Integer	Friend's User ID (Foreign Key referencing Users)

Chats Table

Column	Data Type	Description
chat_id (PK)	Integer	Primary Key for Chat
user_id (FK)	Integer	User's ID (Foreign Key referencing Users)
friend_user_id	Integer	Friend's User ID (Foreign Key referencing Users)
message_id (PK)	Integer	Primary Key for Message
message_text	Text	Text of the Chat Message
timestamp	Timestamp	Timestamp of when the message was sent

Currencies Table

Column	Data Type	Description
currency_id (PK)	Integer	Primary Key for Currency
currency_name	String	Name of the Currency
symbol	String	Currency Symbol

Locations Table

Column	Data Type	Description
location_id (PK)	Integer	Primary Key for Location
location_name	String	Name of the Location

Flights Table

Column	Data Type	Description
flight_id (PK)	Integer	Primary Key for Flight
location_id (FK)	Integer	Location ID (Foreign Key referencing Locations)
cost	Decimal	Cost of the Flight
departure_time	Timestamp	Departure Time of the Flight
arrival_time	Timestamp	Arrival Time of the Flight
picture	String (URL)	URL to Flight Image

Hotels Table

Column	Data Type	Description
hotel_id (PK)	Integer	Primary Key for Hotel
location_id (FK)	Integer	Location ID (Foreign Key referencing Locations)
cost	Decimal	Cost of the Hotel
check_in_date	Date	Check-in Date
check_out_date	Date	Check-out Date
picture	String (URL)	URL to Hotel Image

Events Table

Column	Data Type	Description
event_id (PK)	Integer	Primary Key for Event
location_id (FK)	Integer	Location ID (Foreign Key referencing Locations)
cost	Decimal	Cost of the Event
address	String	Address of the Event
city	String	City where the Event is located
zip_code	String	Zip Code of the Event
event_time	Timestamp	Time of the Event
picture	String (URL)	URL to Event Image

Itineraries Table

Column	Data Type	Description
itinerary_id (PK)	Integer	Primary Key for Itinerary
user_id (FK)	Integer	User's ID (Foreign Key referencing Users)
location_id (FK)	Integer	Location ID (Foreign Key referencing Locations)
itinerary_name	String	Name of the Itinerary

shared_with	String	Comma-separated list of user_ids for sharing

Blogs Table

Column	Data Type	Description
post_id (PK)	Integer	Primary Key for Blog Post
user_id (FK)	Integer	User's ID (Foreign Key referencing Users)
location_id (FK)	Integer	Location ID (Foreign Key referencing Locations)
post_text	Text	Text content of the Blog Post
replies	String	Comma-separated list of reply_ids

Add/Delete/Search architecture:

MongoDB	Add	Delete	Search
Users	New user profiles	User accounts	Other users by username or name
Friends	Friend requests	Unfriend users	-
Chats	Chat messages	Own chat messages	Chat messages based on sender, recipient, or timestamp
Currencies	New currencies	Currency removal	-
Locations	New locations	Location removal	Location by name and zip_code
Flights	Flight details	Remove flights from itineraries	Flights based on flight_id, location, cost, and time
Hotels	Hotel details	Remove hotels from itineraries	Hotels based on hotel_id, location, cost, and date
Events	Event details	Remove events from itineraries	Events based on name, rating, location, cost, city, zip_code, and time
Itineraries	New itineraries	Own itineraries	Own itineraries and shared itineraries based on itinerary_id, location and user_id
Blogs	New blog posts	Own blog posts	Blog posts based on blog_id, rating, location_id, user_id, and other criteria

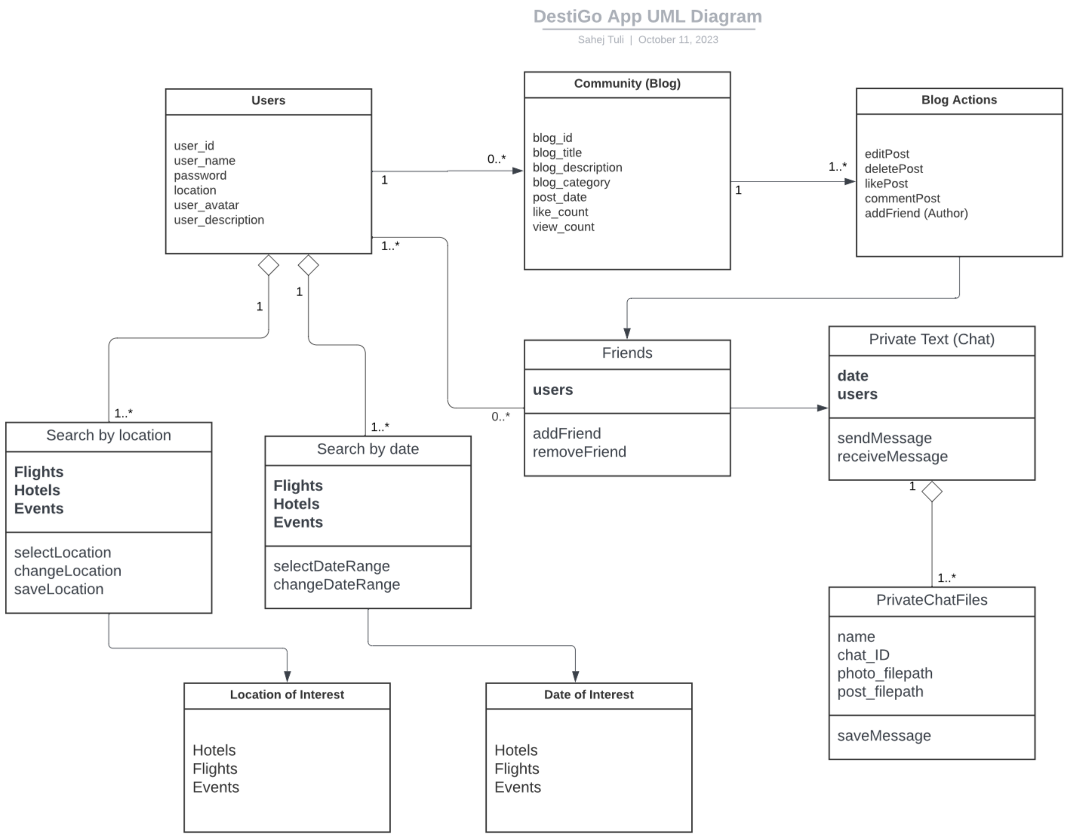
DB operations:
In Mongodb we can use queries to obtain the data we want. For example, we can filter our data in the user's namespace by typing { "username": "Sahej" } which shows us all users with the name Sahej.

Own Api:
- Users, Friends, Chats, Blogs, Itinerary(location)
- Our API is created using express and uses async/await calls(mongoose) to do add/delete/search.
- We also use JSON web tokens to encrypt the data we get from users.

3rd Party API:

- Itineraries (Flights, Hotels, Events)
- For Hotels we are going to use an API created by Api Dojo, they use Axios, and have data on location, cost, and name.
- For Flights we are going to use an API created by Travelpayouts, they use Axios and have data on cost (cheapest tickets), name, and popular airlines,
- For Events we are going to use an API created by OpenWeb Ninja, which use Axios and have data on location and date.

5. High Level UML Diagram



6. Identify actual key risks for your project at this time

- Skills Risks and Mitigation Plan
  - If a team member does not understand how to do their task they should:
    - ask for help in the Discord
    - attend office hours
    - use resources that were recommended in class
  - If there are in emergencies and a team member doesn't finish their study plan then:
    - start studying earlier
    - don't procrastinate
    - plan ahead of time for emergencies

- Schedule Risks
  - If changes are made to any part of the code and others weren't notified then:
    - update Discord immediately
    - communicate with the team before pushing the changes
    - create a pull request
  - If a team member doesn't finish their task on time then:
    - ask for help in advance
    - start earlier
    - attend office hours in advance
- Teamwork Risks
  - If team members don't understand what to work on because they were not assigned a task then:
    - take initiative and let the team know what you want to work on and get confirmation before doing it
    - communicate with the team lead and ask what they need help on
  - If the code style is so different to the point where other backend/frontend leads don't understand it then:
    - add comments to explain your code
    - match coding styles so that it is easier for others to comprehend your code
- Legal/Content Risks
  - If a team member is unsure if an addition to the website is a potential risk then:
    - consult with the professor
    - consult with the team
    - research to learn more

7. Project management

Outside of class, we schedule meetings over Zoom which are recorded for team members to watch if they aren't able to attend. During each meeting, tasks that need to be completed are discussed along with the progress on tasks that were assigned during the previous meeting. Our team leader takes point on the discussion, checks on our progress, and proposes our next steps. We are also active in our Discord server outside of meetings by asking questions and updating each other on our progress in real-time.

To manage tasks, we use Discord threads made by our team leader in which each member is assigned duties in accordance with their role when a new milestone is introduced. Each thread is appropriately named with a due date and is used as a specific channel to talk about the current tasks. If a team member is struggling with a task or falls behind, we use Discord to ask for help immediately so that we can delegate another person to work with them to complete the task. In the event a task is not finished, we can have a discussion on what we can do as a team to help them, figure out why it happened, and complete the task as soon as possible.

Team lead ensures that all members have read and understood this document.	Yes, 10/11/2023 – SAHEJ TULI