

# Activity Tracker

Life is full of various kinds of events. Sometimes, it can feel impossible to keep track of everything. Activiti aims to help the user take control of their life, by helping them keep track of events, exercise, and more.

Github: <https://github.com/nasser-csumb/activiti>

Youtube: <https://youtu.be/9unG8LPMHmo>

## Table of contents

<b>Completed Rubric Items</b>	<b>2</b>
<b>Initial Layout</b>	<b>3</b>
Entity Relationship Diagram (ERD) REQUIRED	4
<b>Team Members:</b>	<b>5</b>
<b>Jian Mitchell</b>	<b>5</b>
<b>Nasser Akhter</b>	<b>5</b>
<b>Noah deFer</b>	<b>6</b>
<b>Suhaib Peracha</b>	<b>7</b>
<b>User Stories</b>	<b>8</b>
User Stories (Noah deFer)	8
Exercise User Stories (Suhaib Peracha)	9
Wellness User Stories (Nasser Akhter)	9
Travel & Exploration User Stories (Jian Mitchell)	9

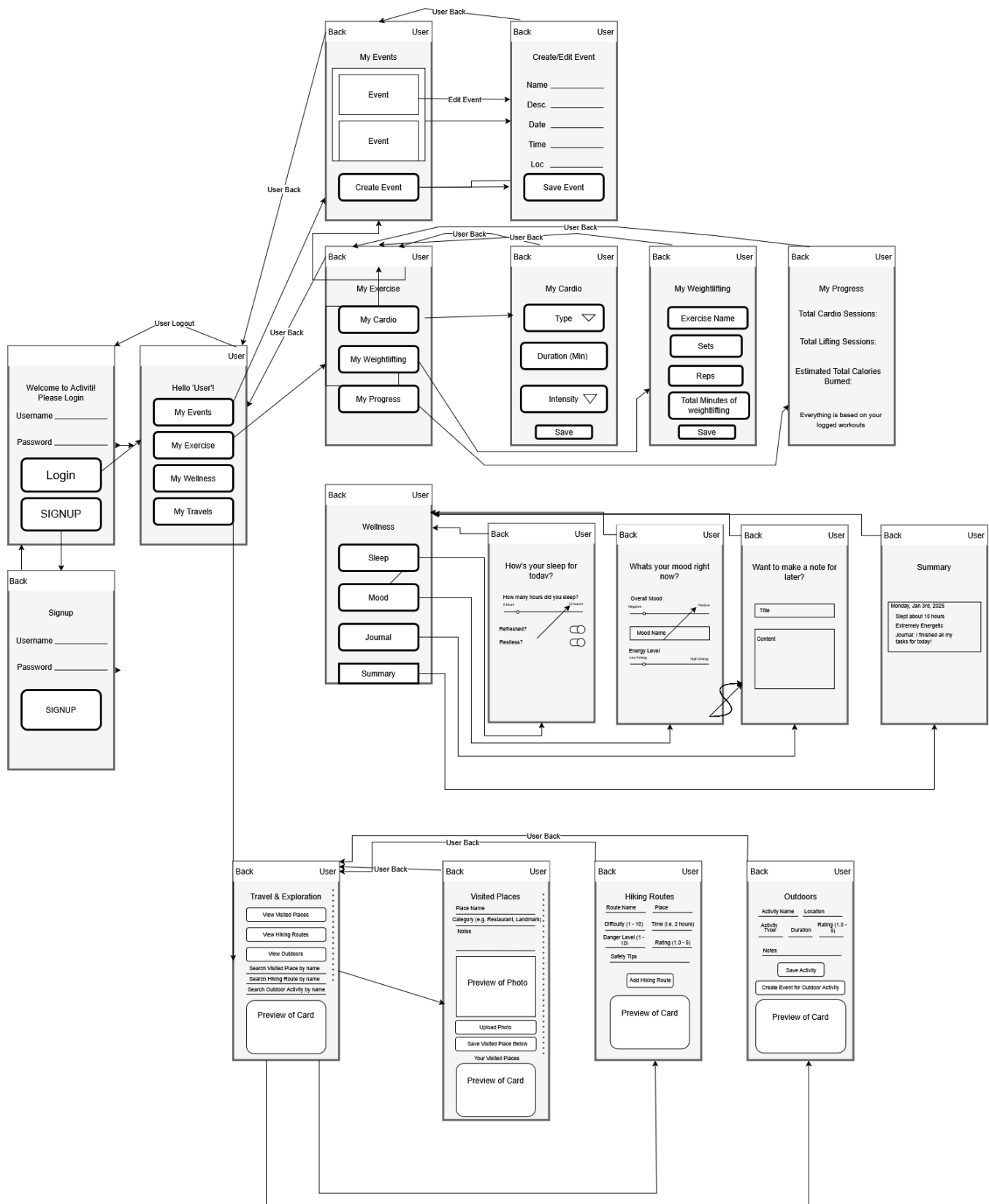
## Completed Rubric Items

Please **Highlight** the items you have completed

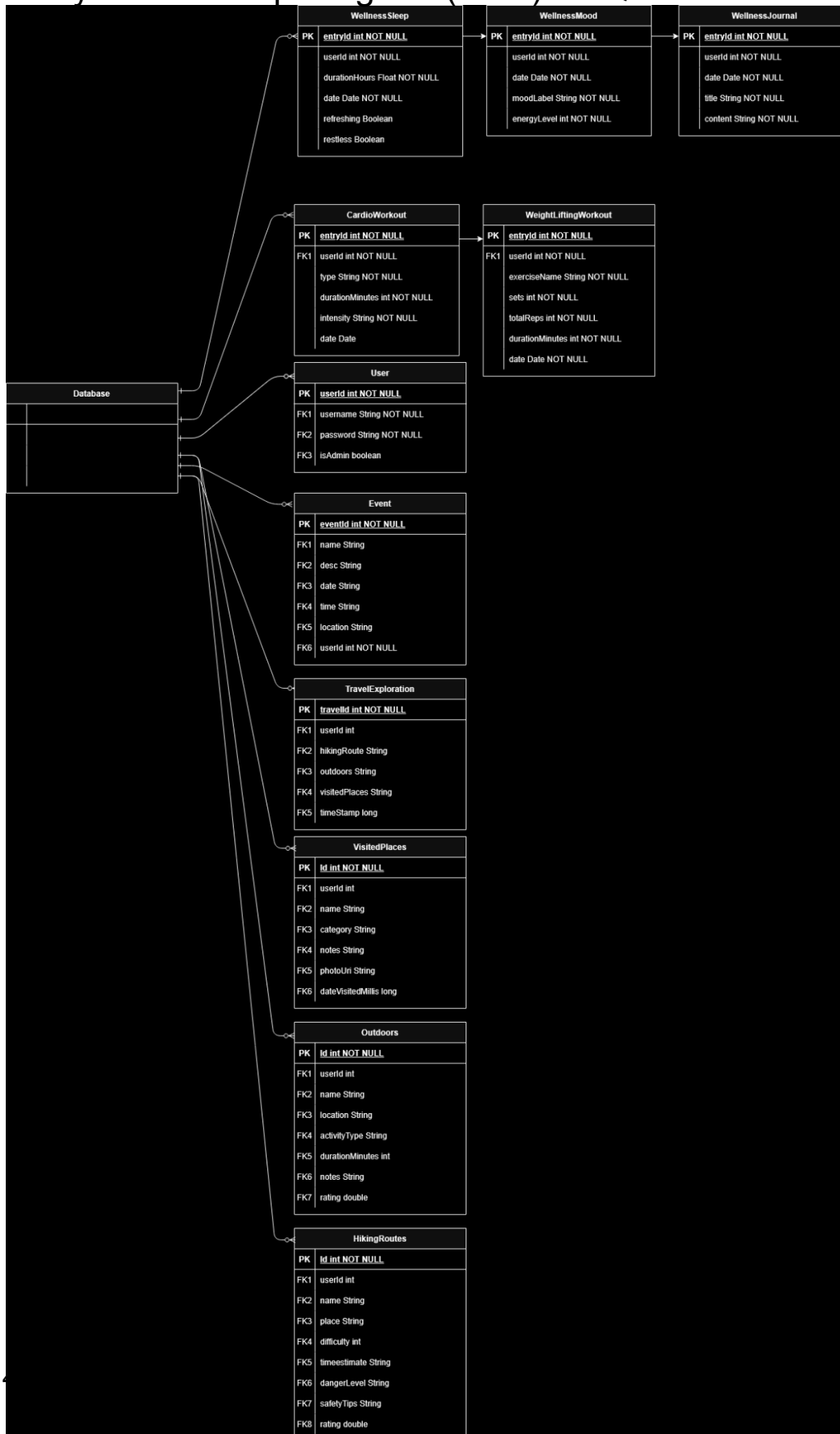
Item	The following are Required	Score
<b>Persistence</b>	Minimum 3 tables implemented using the Room database wrapper If you do not use the Room Database wrapper the project will be marked as 0.	30
<b>Github issues</b>	There must be at least 5 GitHub issues PER TEAMMATE	10
<b>Presentation</b>	Each application must have a video showing its functionality. The video must be between 6 – 8 minutes long. Each person must present their portion for 1-2 minutes	35
<b>Highlighted Rubric</b>	This rubric must be included in the video AND the submission. All completed tasks must be <b>highlighted</b>	5
<b>Activities</b>	Each team member must create 2-3 activities, one of which must interact with the database.	12
<b>Activity: Login Page</b>	An activity where a username and password are entered. The username and password must be stored in the local room database.	4
<b>Activity: Landing Page</b>	After successful login an activity should start that displays the username and some other information retrieved from the local database.	4
<b>Admin landing page</b>	If the user is an administrator <b>the landing page must indicate this</b> with additional options. <b>The same Activity may be used for both admin and non-admin users.</b>	4
<b>Github</b>	Code must be pushed to Github in a <b>public</b> repository. Each teammate must have at least 3 branches, with 5 commits EACH; each branch must be merged into main using a pull request.	10
<b>Unit tests</b>	Each teammate must contribute at LEAST 2 unit tests. The tests must pass.	20

Optional Item	The following are <b>Optional</b>	Score
<b>Intent Factories</b>	All Intents to start an activity must use the factory pattern Either each activity must have an Intent factory method OR use a Factory class.	10
<b>Intent Tests</b>	At least 1 unit test per activity transition. That is each intent must have a unit test.	20
<b>Database Tests</b>	Each database table must have at least <a href="#">three tests</a> (insert, update, delete)	20
<b>Recycler View</b>	At least 1 activity has a <a href="#">recycler view</a> for displaying information	10
<b>Livedata</b>	Incorporate <a href="#">liveData</a> in your project. This dovetails nicely with adding a recycler view	10
<b>API Integration</b>	Use <a href="#">Retrofit</a> to consume an external API	20
<b>External Library</b>	Use an external Library. <a href="#">Here</a> is a list of examples. <b>Retrofit</b> is included and would count for this.	16
<b>Fragments</b>	Incorporate <a href="#">fragments</a> into your application	8
<b>OAuth</b>	Implement login with OAuth 2. This can replace the login procedure described above	16

## Initial Layout



## Entity Relationship Diagram (ERD) REQUIRED



## Team Members:

### Jian Mitchell

- Issue: Implement Activity Type: Travel & Exploration - Visited Places
- Issue: Implement Activity Type: Travel & Exploration - Hiking Routes
- Issue: Implement Activity Type: Travel & Exploration - Outdoors
- Issue: Implement Activity Type: Travel & Exploration
- Issue: Implement Entity Type: TravelExploration
- Issue: Implement Database Type: TravelExplorationDAO
- PR: Merge the Activity Type Visited Places to main
- PR: Merge the Activity Type Hiking Routes to main
- PR: Merge the Activity Type Outdoors to main
- PR: Merge the Activity Type Travel & Exploration to main
- PR: Merge the Entity Type TravelExploration to main
- PR: Merge the Database Type TravelExplorationDAO to main
- Feature: Visited Places: users can keep track of the places they visited and categorize them
- Feature: Visited Places: (optional) interactive map to view the visited places
- Feature: Visited Places: users can upload photos and takes notes of each location
- Feature: Hiking Routes: Safety tips for each hike
- Feature: Hiking Routes: Difficulty ratings and reviews for trails
- Feature: Hiking Routes: (optional) GPS tracking
- Feature: Outdoors: outdoor spots based on user preferences
- Feature: Outdoors: event planning for group outdoor activities
- Feature: Outdoors: (optional) weather alerts for outdoor destinations

### Nasser Akhter

- Issue: Created Repository
- Issue: Wellness Sleep: Create ability to record an entry
- Issue: Wellness Sleep: Create UI
- Issue: Wellness Activity: Add in buttons to sleep, mood, journal, and summary
- Issue: Wellness Activity: Create background image
- Issue: Wellness Activity: Create main activity
- Issue: Wellness Summary: Create UI
- Issue: Wellness Journal: Create ability to record entry
- Issue: Wellness Journal: Create UI
- Issue: Wellness Mood: Create ability to record entry
- Issue: Wellness Mood: Create UI
- PR: Merge the Mental Wellness Activity to main
- PR: Merge the Mental Wellness Sleep activity
- PR: Merge the Mental Wellness Mood Activity
- PR: Merge the Mental Wellness Journal Activity
- PR: Merge related pages such as create entity, update, and delete activities

- Feature: Wellness - User should be able to see four buttons, one for sleep, mood, journal, and summary
- Feature: Wellness - Users should be able to click sleep and be navigated to the Sleep activity.
- Feature: Wellness - Users should be able to click mood and be navigated to the Mood activity.
- Feature: Wellness - Users should be able to click journal and be navigated to the Journal activity.
- Feature: Wellness - Users should be able to click summary and be navigated to the Summary activity.
- Feature: Sleep - Users should be able to see an input for hours slept, refreshed, restless
- Feature: Sleep - Users should be able to interact with the inputs
- Feature: Sleep - Users should be able to save the entry
- Feature: Mood - Users should be able to see inputs
- Feature: Mood - Users should be able to interact with the inputs
- Feature: Mood - Users should be able to save the entry
- Feature: Journal - Users should be able to see inputs
- Feature: Journal - Users should be able to interact with the inputs
- Feature: Journal - Users should be able to save the entry
- Feature: Summary - Users should be able to see the first day (latest) for which entries have been made
- Feature: Summary - Users should be able to scroll between other days with entries

## Noah deFer

- Issue: Implemented Login Activity
- Issue: Implemented Main Menu Activity
- Issue: Implemented Event Activity
- Issue: Implemented Database Functionality
- Issue: Implemented User Table
- Issue: Implemented Signup Activity
- Issue: Implemented Back Button
- Issue: Implemented Logout Button
- Issue: Implemented Event Creation.
- Issue: Add Admin Features to Main Menu
- Issue: Add Recycler to EventActivity.
- Issue: Signup button says that user exists when user does not exist.
- PR: Merged event\_activity into main.
- PR: Merged database\_setup into main.
- PR: Merged login into main.
- PR: Merged noah\_test\_setup into main.
- PR: Fixed bug in SignupActivity.
- PR: Merge event\_recycler into main.
- PR: Fixed admin feature bug, added trim statements to methods.
- Feature: Users can log into the application
- Feature: Users can sign up for the application.
- Feature: Users can log out of the application.

- Feature: Users can navigate to different activities through the main menu
- Feature: Users can add, delete, edit, and view events.
- Feature: Admins have access to extra features in MainActivity.

## Suhaib Peracha

- Issue: Implement Activity Type: Physical Activity – Cardio Workouts
- Issue: Implement Activity Type: Physical Activity – Weight Lifting
- Issue: Implement Activity Type: Physical Activity – Progress Summary
- Issue: Implement entity: CardioWorkout
- Issue: Implement entity: WeightLiftingWorkout
- Issue: Implement database: CardioWorkoutDAO
- Issue: Implement database: WeightLiftingWorkoutDAO
- PR: Merge ActivitiDatabase.java changes into main (Cardio & Lifting entities + DAOs registered)
- PR: Fixed DAO tests for CardioWorkoutDAO and WeightLiftingWorkoutDAO
- PR: Added a Retrofit API integration for motivational quotes
- PR: Integrated Exercise section with database tests
- PR: Added intent-based unit tests for Exercise activities
- PR: Final UI fixes for Exercise section
- PR: Integrated top menu (user + back button) across Exercise activities
- PR: Initial implementation of Exercise activities (Cardio, Weight Lifting, Progress)
- Feature: Planned clean layout for Exercise section: Cardio, Weight Lifting, and Progress
- Feature: Completed UI flow and screen designs for all 3 activities
- Feature: Updated ActivitiDatabase.java with cardio and weightlifting entities and DAOs
- Feature: Matched database structure to UI for simple Room integration (via CardioWorkout and WeightLiftingWorkout tables)

## User Stories

### User Stories (Noah deFer)

Some example user stories:

- As an Activiti user, I want to be able to log in to the application to see my information.
- As an Activiti user, I want to be able to logout of the application.
- As an Activiti user, I want to be able to navigate back to previous activities, such as the main menu.
- As an Activiti user, I want to be able to quickly and easily get to my events, exercises, and travel plans through the main menu.
- As an Activiti user, I want to be able to view and add upcoming events.

From these use cases we generate the following issues:

- Create a Main Menu activity.
  - Using MainActivity?
  - Setup buttons to navigate to other activities.
- Implement a Room database
  - Add the Gradle requirement.
  - Setup the Asynchronous method to call the database.
- Create a Repository class to access the database.
- Create a User Entity object.
  - Required fields: userId, username, password, isAdmin?
  - Create a Room DAO for the User entity.
  - Create methods in the Data repository to access the user records.
- Create a Login Menu activity.
  - Have the LoginActivity pull from the User repository to validate login information.
  - Have the LoginActivity navigate to the main menu when successfully logging in.
- Create a Logout button.
  - Have the logout button clear user login preferences.
  - Have the logout button navigate to LoginActivity.
- Store login information in between activities.
  - Using shared preferences.
- Create an Event Entity object.
  - Required fields: eventId, name, description, date, time, userId.
  - Create a Room DAO for the Event Entity.
  - Create methods in the Data repository to access the event records.
- Create an Events activity.
  - Display events to the user.
  - Allow the user to add, delete, or edit events.
- Create a Back button.
  - Have the Back button navigate back to MainActivity, or the most recent activity.



## Exercise User Stories (Suhaib Peracha)

- As an Activiti user, I want to log my cardio workouts so I can track the type, duration, and intensity.
- As an Activiti user, I want to record my weightlifting sessions by entering the exercise name, sets, reps, and duration.
- As an Activiti user, I want to view a summary of all my exercise activities (Cardio and weightlifting) so I can track my daily progress.
- As an Activiti user, I want a clean and simple interface to access and record exercise data with minimal effort.
- As an Activiti user, I want to see a visual summary of my progress without manually calculating anything.

## Wellness User Stories (Nasser Akhter)

- As an Activiti user, I want to see four wellness options so I can track sleep, mood, journal, and summary.
- As an Activiti user, I want to click on "Sleep" to track how I slept.
- As an Activiti user, I want to click on "Mood" to log how I feel.
- As an Activiti user, I want to click on "Journal" to write my thoughts.
- As an Activiti user, I want to click on "Summary" to review my wellness history.
- As an Activiti user, I want to input how many hours I slept and how I felt during sleep.
- As an Activiti user, I want to save my sleep entry for future reference.
- As an Activiti user, I want to rate or describe my mood for the day.
- As an Activiti user, I want to save my mood entry to track emotional trends.
- As an Activiti user, I want to write in a journal input field.
- As an Activiti user, I want to save my journal entry.
- As an Activiti user, I want to see a summary of my latest wellness data.
- As an Activiti user, I want to scroll through previous wellness entries by day.

## Travel & Exploration User Stories (Jian Mitchell)

- As an Activiti user, I want to be able to search for destinations and get recommendations on my interests.
- As an Activiti user, I want to be able to save and organize my travel for upcoming trips.
- As an Activiti user, I want to be able to track my past explorations and review travel memories in a timeline.
- As an Activiti user, I want to be able to discover local attractions near my destination.
- As an Activiti user, I want to be able to find nearby hiking trails, with difficulty levels, and time.
- As an Activiti user, I want to be able to share my hiking experiences with other users.
- As an Activiti user, I want to be able to plan camping trips, which includes the location, gear, and weather.
- As an Activiti user, I want to be able to join outdoor community events, such as group hikes or conservation activities.

- As an Activiti user, I want to be able to log experiences during explorations.