

# **IIT Off-Campus Escorts Application**

## **CS-487 Software Engineering**

### **Illinois Institute of Technology**

Jiranun Jiratrakanvong A20337992

Shreyas Moudgalya A20321695

Karthik Vasudevan A20330010

#### **Problem Statement**

Illinois Institute of Technology continuously strives to provide services designed to keep our campus safe. In support of this effort, the IIT Public Safety Department (IIT PSD) offers security on-campus and off-campus escorts to IIT faculty, staff, and students in situations where individuals have a significant concern about their personal safety. For the off-campus escorts, they are available between the hours of 7 p.m.–3 a.m.[1]

However, the van escorts only begin from a central pick-up point which is Galvin Library and are offered on a first-come, first-served basis. Passengers will be dropped off at only five off-campus locations. Because it's all about safety, we think that there should be more than one pick-up location, and more than five drop-off locations. It would be more convenient if passengers are able to reserve a seat on the van any round trip they want, and drivers are able to know where they have to go for picking up and dropping off by using their phone. Moreover, passengers will also know their exactly round trip after making a reservation instead of going to Galvin Library, and waiting in line. Consequently, IIT Off-Campus Escorts Application will be the application solving these problems.

#### **User Analysis and Requirements spec**

- **Categories of users - Target Audience**
  - Passengers
    - The end users of this product would be any current student of Illinois Institute of Technology. The students include full time Registered Undergraduate students and Graduate students.
  - Drivers
    - The CMS System admins would be the Public Safety Personnel. Multiples Public Safety Personnel can control the application.

- **Functional Requirements**

- The App must cater to 2 sets of users. Passengers and Drivers.
- The App must be able to register a ride for a given passenger.
- The App must send a confirmation of the requested ride after booking.
- The App must be able to display the pickup locations of Passengers for the Driver.
- Users can cancel the trip before the trip began 30 mins.
- Drivers can see the navigation route to all locations they have to drive to.
- Interface requirements
  - Time Slot booking button accepts touch entry.
  - Time Slot booking button is activated only 2 hours before the booking time.
- Business Requirements
  - Booking must be confirmed before a request can be approved.
  - Clicking the Confirm button moves the request to the Approval Workflow.
- Regulatory/Compliance Requirements
  - The database will have a functional audit trail.
  - The system will limit access to authorized users.
  - The spreadsheet can secure data with electronic signatures.

- **Non-Functional Requirements:** Largely met through architecture design. Trade-offs are necessary due to conflicting priorities and/or overlaps in design elements.

- Accessibility Requirements: The app shall provide help text in English. Data: The app shall collect student ID
- Auditing and Reporting Requirements: A record of which users access or try to access the app is required.
- Availability Requirements: Functionality “Reserve Seat” is available 06:00pm to 10:00pm daily excluding Weekends and Public Holidays.
- Backup and Recovery: The app can be made available after unplanned system downtime within 4 hours. Data: All user data will be backed-up daily.
- Capacity Requirements: 1 user can reserve only 1 seat per night.
- Compatibility Requirements: The app must be compatible to all versions of iOS. Data: Customer data can be exported in XML format.
- Concurrency Requirements: Up to 100 users may be using “Request Ride” at any one time.
- Error-Handling Requirements: In the event of the user cancelling or quitting the process “Request Ride” any changes made by the user will be reversed.
- Legal and Regulatory Requirements: All changes to Customer data will be held for 6 years from the date of change.

- Localizability Requirements: For American users of “Update Customer” all dates will be displayed in MM/DD/YY format.
- Performance Requirements: During the process “Request Ride” system responses should be no more than 5 seconds.
- Precision Requirements: Time of changes to data must be recorded to the nearest second.
- Reliability Requirements: “Request Ride” will be available to users 98% of normal working hours.
- Scalability Requirements: The platform supports up to 20,000 users.
- Security Requirements: Only users holding the role “Passenger” can access “Request Ride”.
- Stress Requirements: Up to 10 Public Safety Driver users may access the Passenger details.
- Supportability Requirements: There should be a “Get Help” feature for the passenger to leave message or feedback.
- Ease of Use Requirements: The passenger must be able
- Security Requirements: The app should implement a user login and must store each user details.

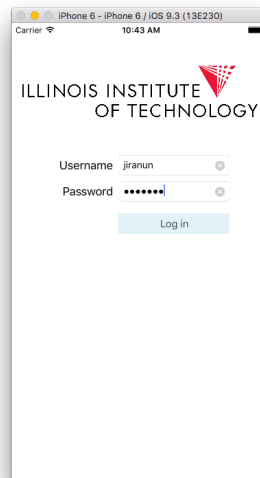
- **Environmental Requirements**

- Runs as an app on smartphone.

## Design

### UI Design

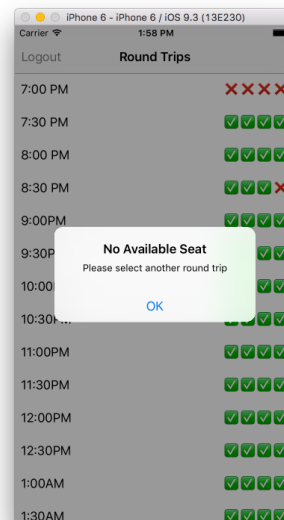
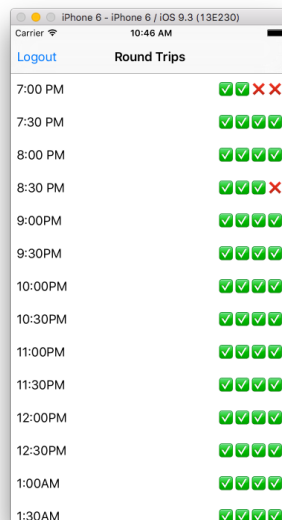
The UI is separated into two types based on user's types (Passenger or Driver). The system will know user's type automatically by logging in using an IIT account.



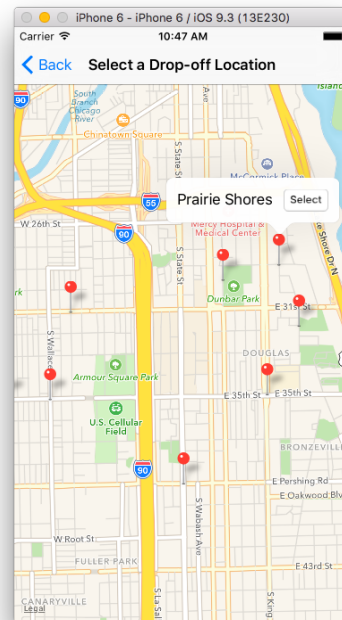
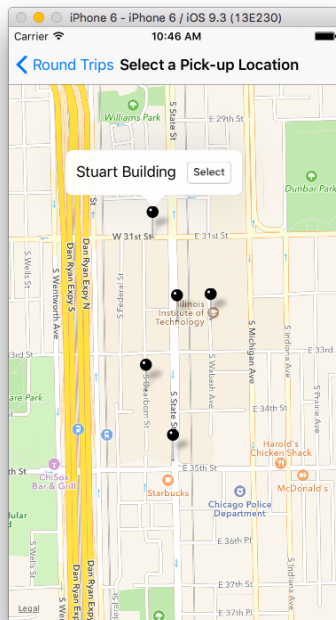
### 1. Passengers

After logging in, users will be able to choose a round trip, a pick-up location, a drop-off location, and confirm or cancel the trip.

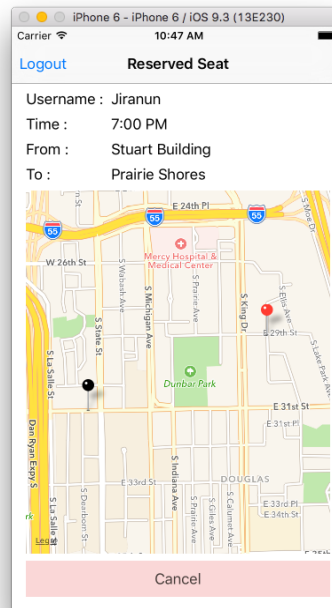
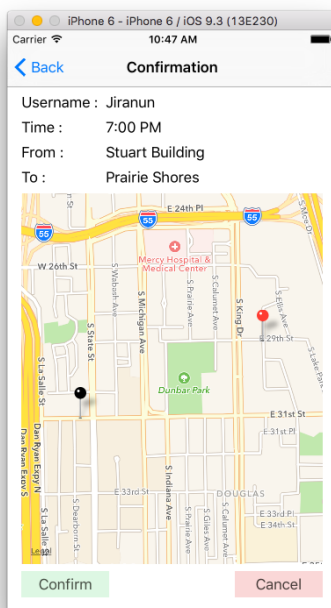
For choosing a round trip, the UI is going to be a table of round trips, so users can just tap on the round trip they want. Users can also see how many seats are available. If a user selects the round trip which is already full, there's a pop-up message asking the user to choose another round trip.



For choosing a pick-up location or drop-off location, the map will be showed with location pins on it. Users can tap on the pin to see the name of location, and select the location by tapping “Select”



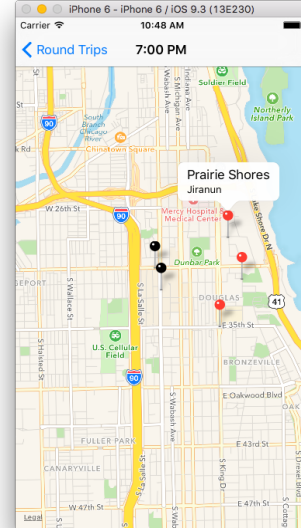
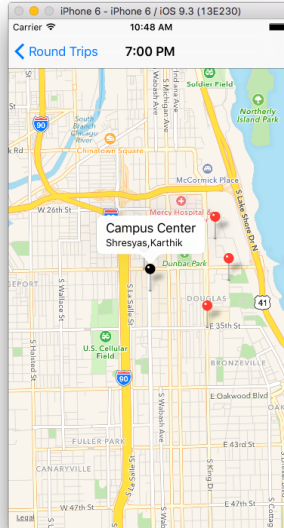
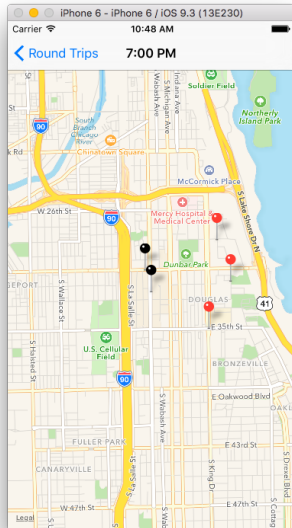
For confirmation or cancellation, users will see all necessary information by text including name, round trip, pick-up location, drop-off location, and also the corresponding location pins on maps.



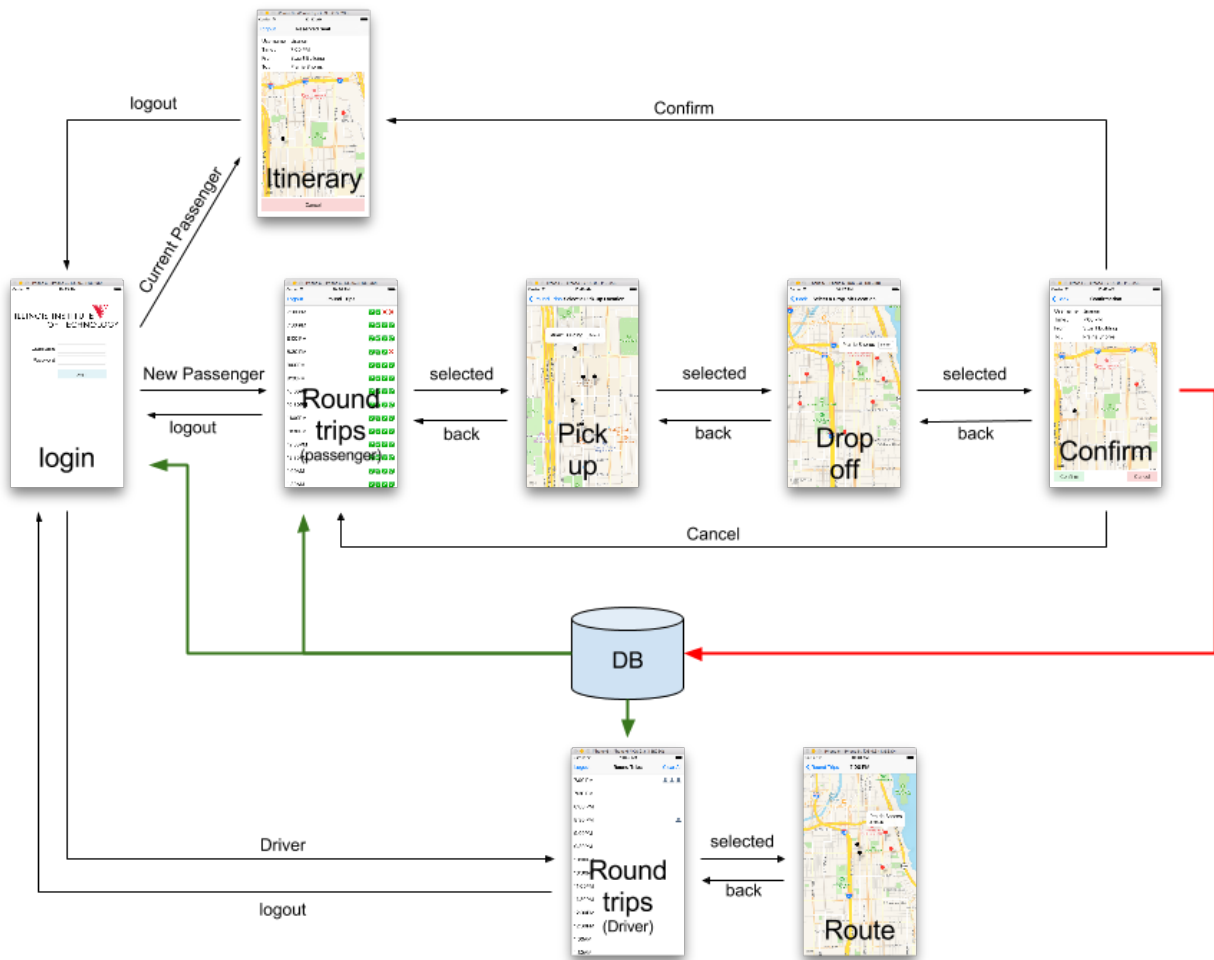
## 2. Drivers

After logging in, drivers will be able to choose a round trip, and see locations where they have to drive to.

For choosing a round trip, the UI is going to be a table of round trips as same as passenger's UI, but instead of seeing available seats, users will see how many passengers will be in the van escorts.

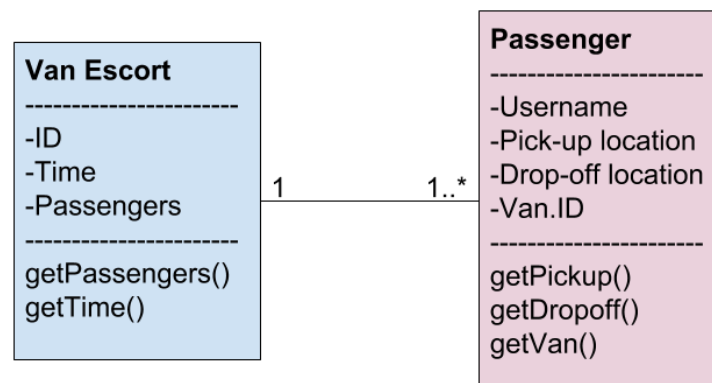


## State-transition Diagram with UI



Click [here](#) for better resolution

## Class and Association Diagram



## **Test Plan**

### **Traceability**

A developed software should be aligned with the proposed requirements to achieve traceability. In this project we use the Requirement traceability matrix, as we develop the software we check the completion of a requirement. A sample traceability matrix is shown below.

Requirement Identifiers	Func_Req_1 1.1	Func_Req_2 1.2	Func_Req_3 1.3	Non_Func_Req_1 2.1	Non_Func_Req_2 2.2	Non_Func_Req_3 2.3	Non_Func_Req_4 2.4	Non_Func_Req_5 2.5	Non_Func_Req_6 2.6	Non_Func_Req_7 2.7
Test Cases										
1.1.1	X									
1.1.2	X									
1.2.1		X								
1.2.2		X								
1.3.1			X							
1.3.2			X							
2.1.1				X						
2.2.1					X					
2.5.1								X		
2.4.1							X			

As we move further into the application, we develop test cases to test the remaining requirements as specified above.

### **Scenarios and use cases**

#### **Scenario 1. User books a trip**

- User logs into the system and reserves a round trip with pickup and drop off locations for a time at his convenience.
- User logs into the system and tries to reserve a round trip for a closed slot. He proceeds with the next available timing.
- User logs into the system and sees his desired time slot is already full. He logs out of the portal.
- User logs into the system and reserves a round trip with pickup and drop off locations for a time at his convenience. He later cancels it 30 minutes prior to departure.
- User logs into the system and tries to reserve a round trip with pickup and drop off locations for a time at his convenience. He later tries to cancel it when less than 30 minutes for departure.
- User logs into the system. Selects an available time slot, sees the pickup and drop off locations, and cancels the trip.
- User Logs in and logs out.



### Scenario 2. Driver chooses a route

- Driver logs into the system and selects a round trip. He sees the number of passengers travelling and navigates to the map showing pick up and drop off locations.
- Driver Logs in, sees the available round trips and logs out.

### **Usability**

- The application is to be analyzed by a small subset of users, and feedbacks about ease of use, unidentified bugs, performance issues will be recorded/incorporated.
- Completion Rate - Completion rate is the percentage of test participants who successfully complete the task without critical errors. A critical error is defined as an error that results in an incorrect or incomplete outcome
- Based on the completion rate, the problem can be prioritized and classified as High, Low or Medium.
- Significant reworks will be carried out based on the problem priority.
- Results - The Usability Test Report will be provided at the conclusion of the usability test. It will consist of a report and/or a presentation of the results; The usability testing will only complete when the maximum possible quality is achieved.

### **Exception Handling**

- The application system should recognize all possible error conditions. The code should not break in any case.
- The unit testing proved that all data types fall in their boundaries( ex: int, float ).
- Improper navigation by the user/driver is handled accordingly.
- Shutting down the application abruptly will maintain a state of the user.
- All exceptions can be tested only when the application is completely built.

### **Conclusion**

The project was successfully completed as per the initial requirements. There were no major changes in the design as we progressed. We tested our application and made sure that it provides maximum quality and guarantees each requirement as specified. We implemented and learned the best practices of the software engineering through this project.

### **References**

1. IIT Public Safety Department  
<http://web.iit.edu/public-safety/safety-awareness/public-safety-escorts>
2. <http://istqbexamcertification.com/what-is-usability-testing-in-software-and-its-benefits-to-end-user/>