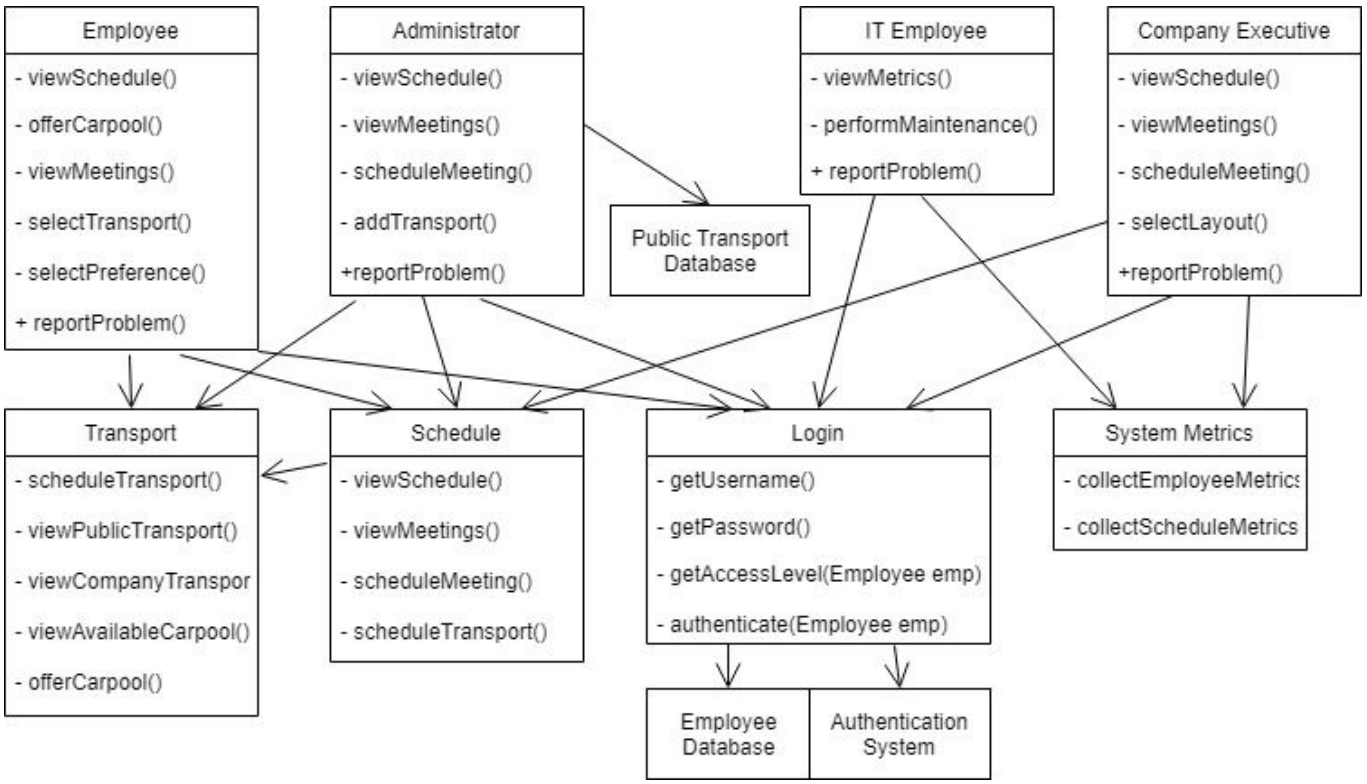
Team Assignment #2:
Design and Specification of MCSTS
Part 1

Jack Sparrow, Reagan Leonard, Jackson Lee
(The Brogrammers)
Dr. Murali Sitaraman
CPSC 3720-002
12 March 2019

Class Diagram(s)



Classes: High Level Descriptions

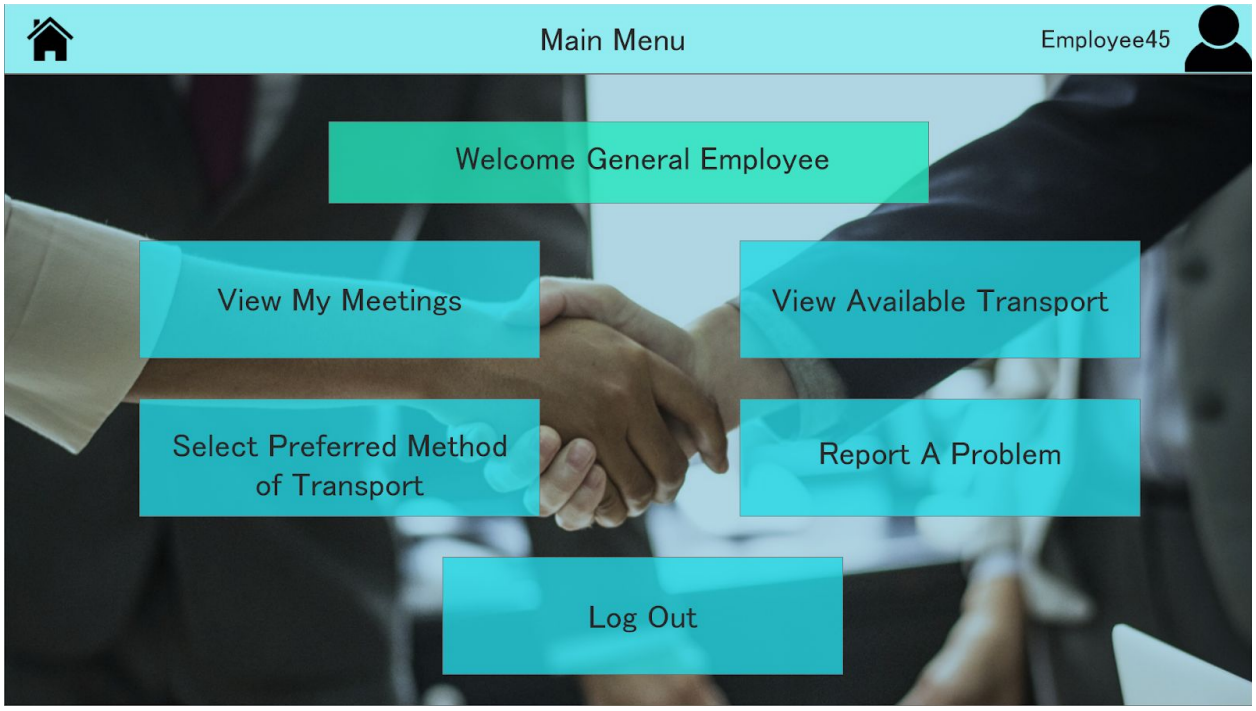
- Employee
    - Description: *This class represents the employee actor that will be using the system. This class should have access to the Schedule, Transport, and Login classes. It should have the following functionality: view schedule, offer carpool, select transport, select preference, and report problem.*
    - Have access to:
        - Schedule Class
        - Transport Class
        - Login Class
    - Functions:
        - viewSchedule();
        - offerCarpool();
        - viewMeetings()
        - selectTransport();
        - selectPreference();
        - reportProblem();
- Administrator
    - Description: *This class represents the administrator actor that will be using and modifying the system. This class should have access to the Schedule, Transport, and Login classes. It should also have access to an external Public Transport schedule. It should have the following functionality: view schedule, edit schedule, and report problem.*
    - Have access to:
        - Schedule Class
        - Transport Class
        - Login Class
        - (External) Public Transport Schedule(s)
    - Functions:
        - viewSchedule()
        - viewMeetings()
        - scheduleMeeting()
        - addTransport()
        - reportProblem()
- IT employee
    - Description: *This class represents the IT employee actor that will be using and modifying the system. This class should have access to the System Metrics and Login classes. It should have the following functionality: view metrics, perform maintenance, and report problem.*
    - Have access to:
        - System Metrics Class
        - Login Class
    - Functions:
        - viewMetrics()
        - performMaintenance()
        - reportProblem()

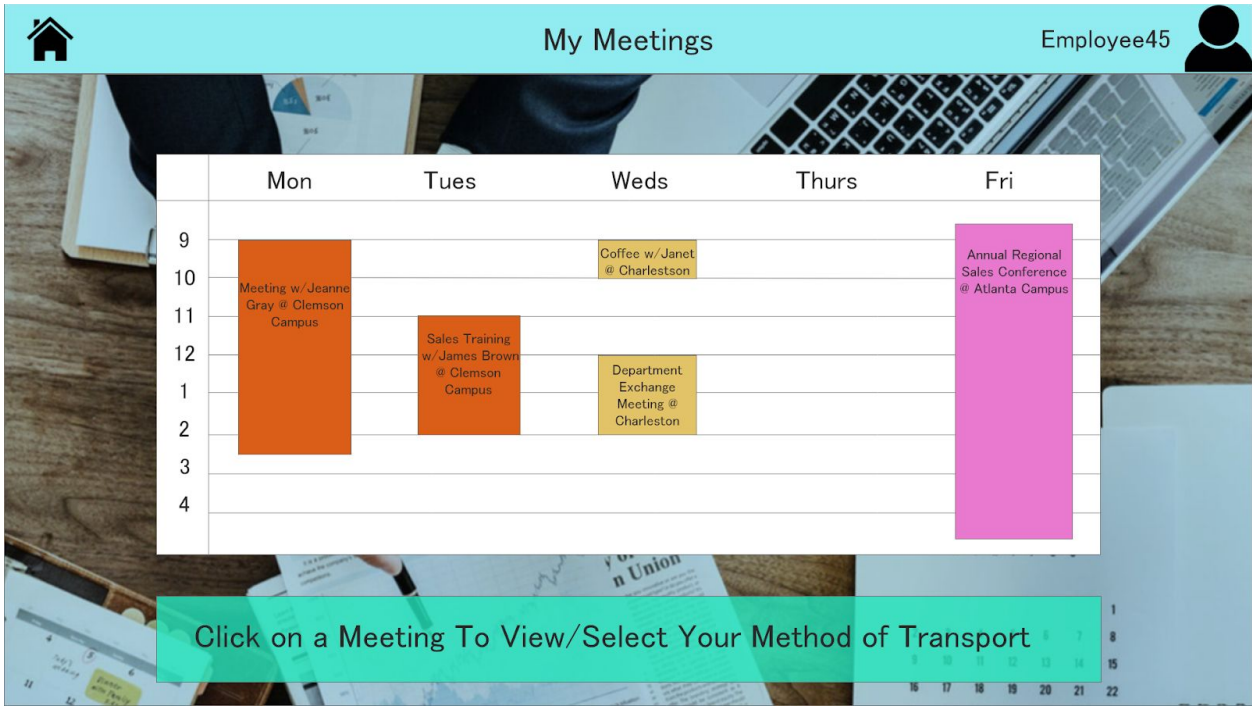
- Company Executive
    - Description: *This class represents the company executive actor that will be using and modifying the system. This class should have access to the Schedule, System Metrics, and Login classes. It should have the following functionality: view schedule, view meetings, schedule meeting, view metrics, select system layout, and report problem.*
    - Have access to:
        - Schedule Class
        - System Metrics Class
        - Login Class
    - Functions:
        - viewSchedule();
        - viewMeetings()
        - scheduleMeeting()
        - selectLayout()
        - reportProblem()
- Transport
    - Description: *This class represents the transport that employees will be able to view and select within the system. This class is accessed by other classes. It should have the following functionality: schedule transport for meeting, view public transport schedule, view company transport schedule, view available carpool, and offer carpool.*
    - Functions:
        - scheduleTransport();
        - viewPublicTransport();
        - viewCompanyTransport();
        - viewAvailableCarpool();
        - offerCarpool();
- Schedule
    - Description: *This class represents the schedule/scheduling employees, administrators, and company executives will be able to view and select within the system. This class should have access to the Transport class. It should have the following functionality: view schedule, view meetings, schedule meeting, and schedule transport for meeting.*
    - Have Access to:
        - Transport Class
    - Functions:
        - viewSchedule();
        - viewMeetings();
        - scheduleMeeting();
        - scheduleTransport();
- Login
    - Description: *This class represents the login function used by the system to authenticate users and classify what level of access each of them have depending on their employment level. This class should have access to the Employee Database and Authentication system. It should have the following functionality: enter username, enter password, forgot password, forgot username.*
    - Have access to:

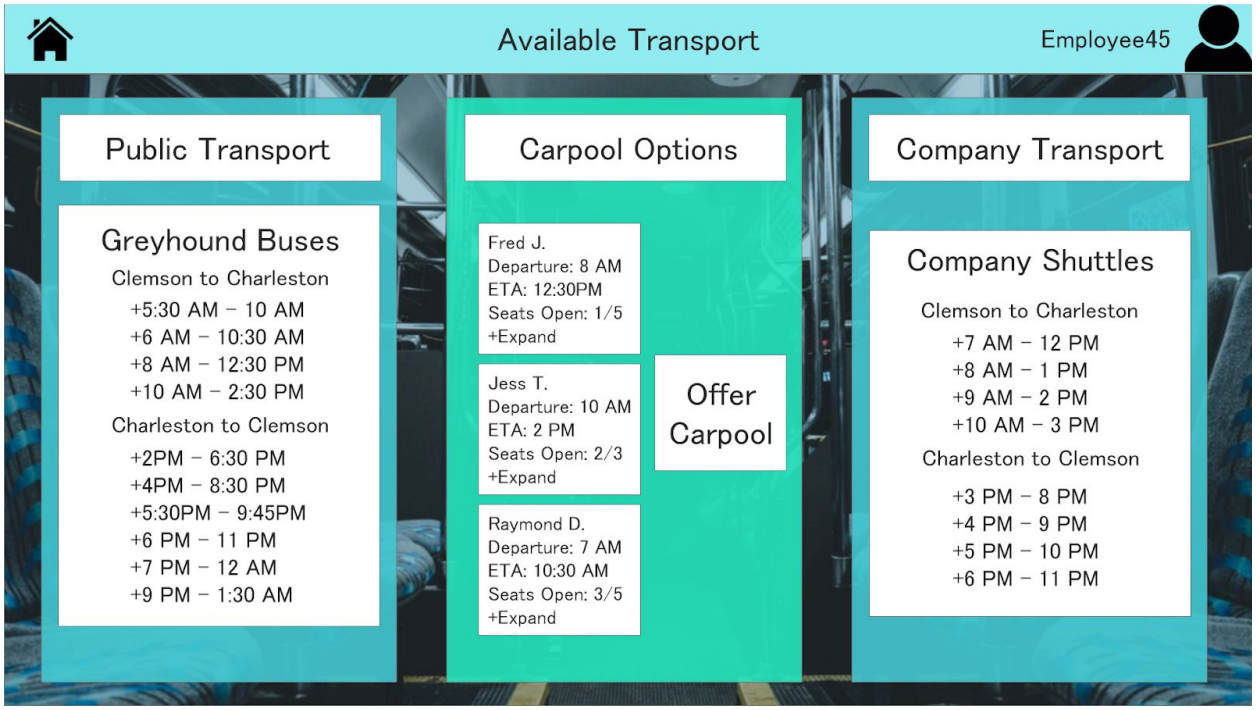
        - Employee Database
        - Authentication System
    - Functions:
        - getUsername();
        - getPassword();
        - getAccessLevel(Employee emp);
        - authenticate(Employee emp);
- System Metrics
    - Description: *This class represents the system's collection of various metrics including but not limited to transport usage data and reported problems. This class should have access to all other classes. It should have the following funtionality: collect employee/schedule/etcetera metrics, display metrics graphically, and display reported error submissions.*
    - Functions
        - collectEmployeeMetrics()
        - collectScheduleMetrics()
        - Collect….
        - ...
- External classes:
    - Employee Database
        - Description: *This is the company database containing all employee information.*
    - Authentication System
        - Description: *This is the authentication system that stores usernames and passwords for a single company.*

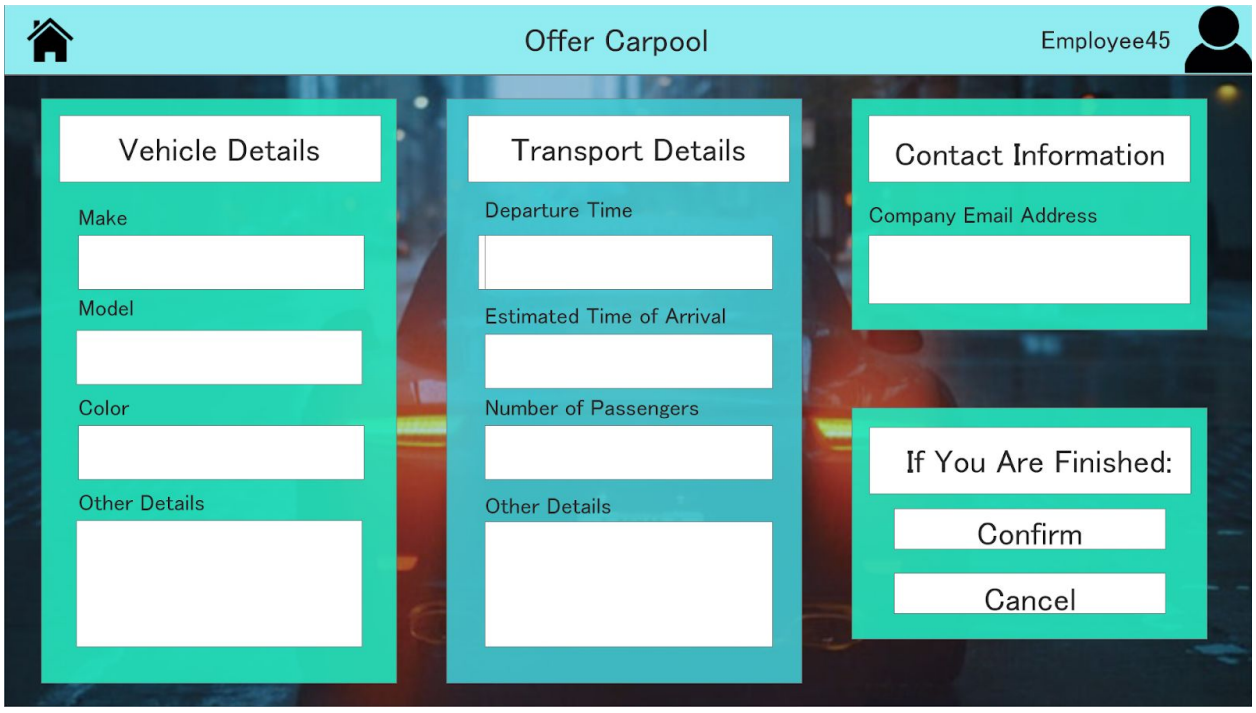
Design Verification: UI Sample

What a General Employee would see in terms of how and where transportations take place:





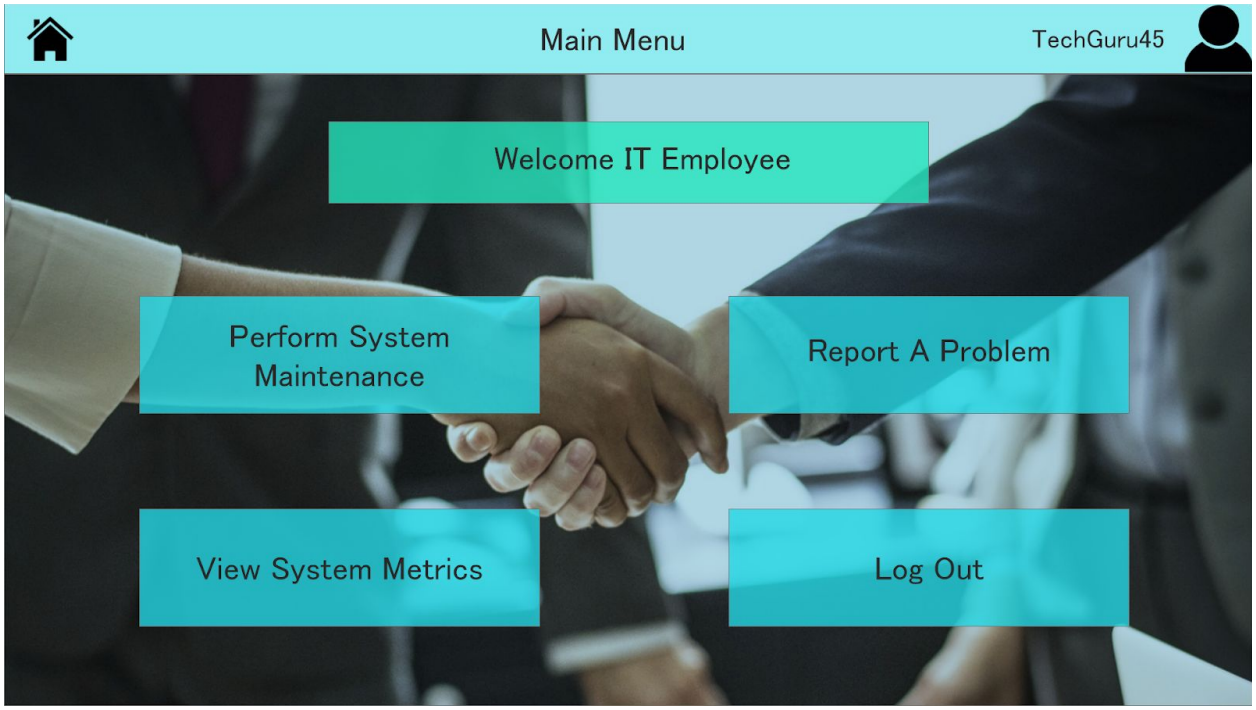


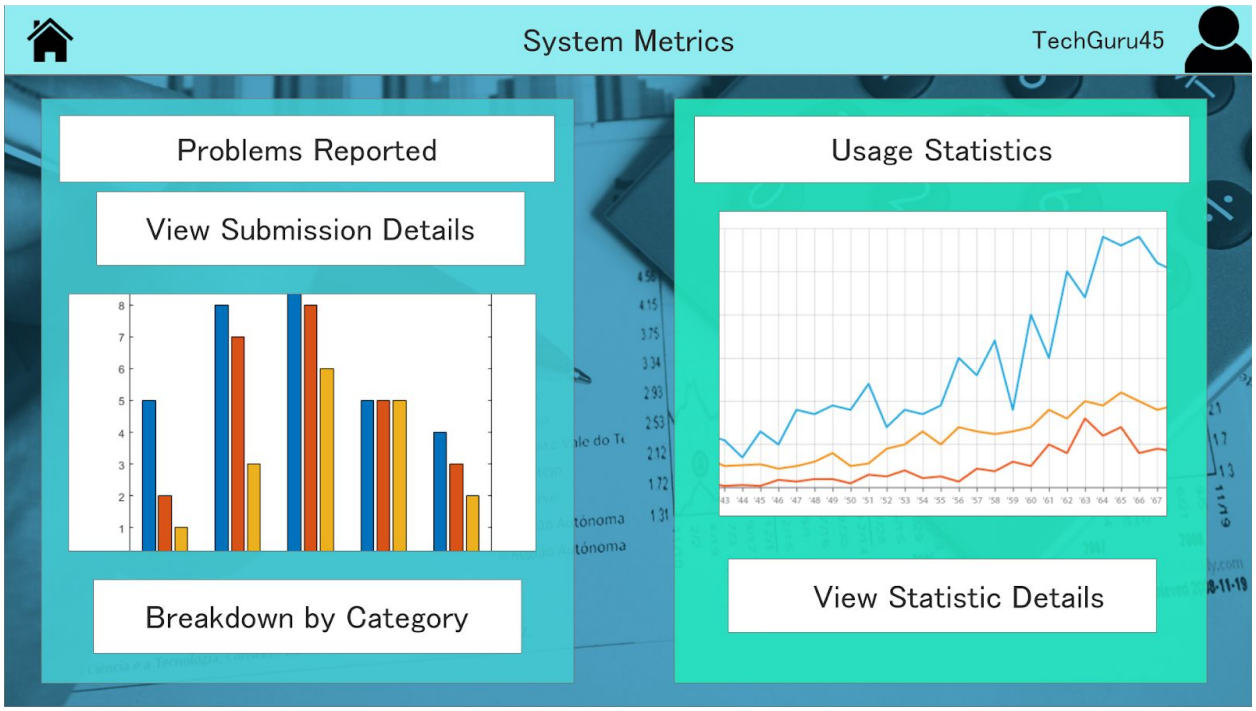


Explanation:

The portion of the system that holds the most important features for general employees is the section dealing with the scheduling of transportation and carpool in relation to a meeting. These preliminary visuals match scenarios outlined within the scenario portion of therequirements documentation. The overall layout matches the UML diagram in relation to what is accessible in each page.

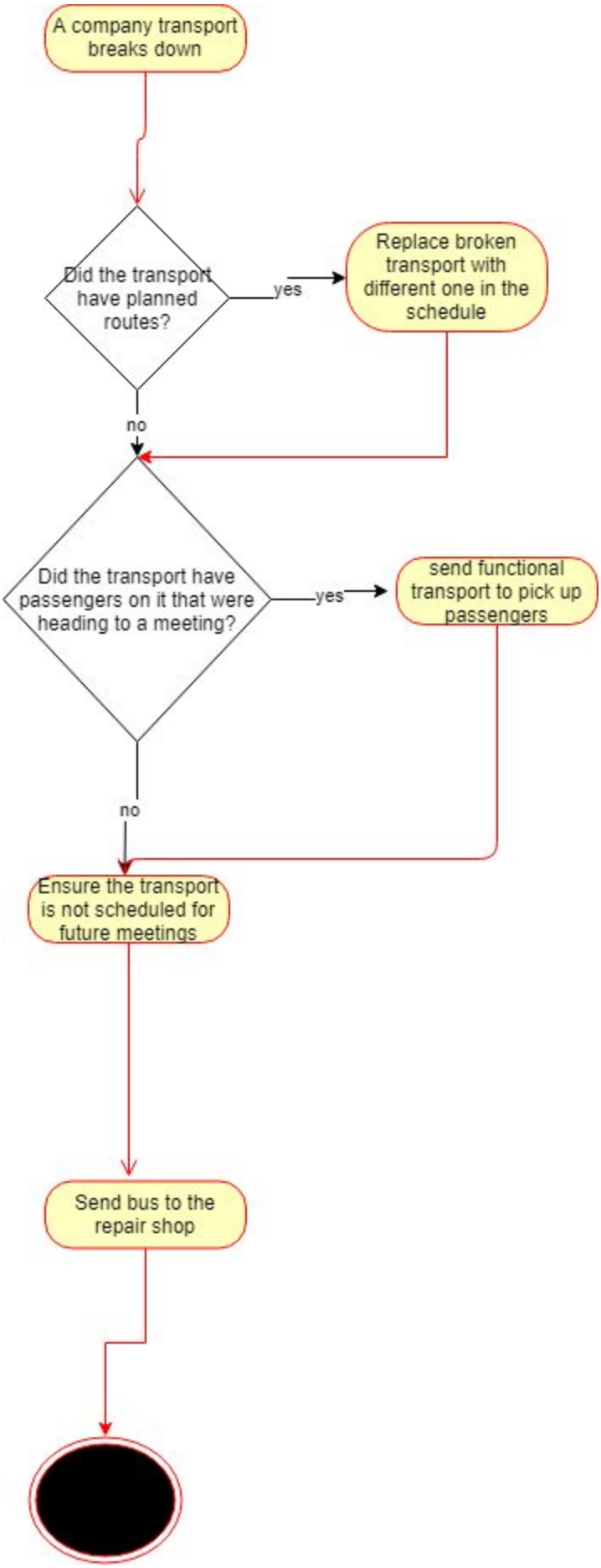
What an IT Employee (someone who manages the system) would see in terms of where the demands of the system and transportation are:

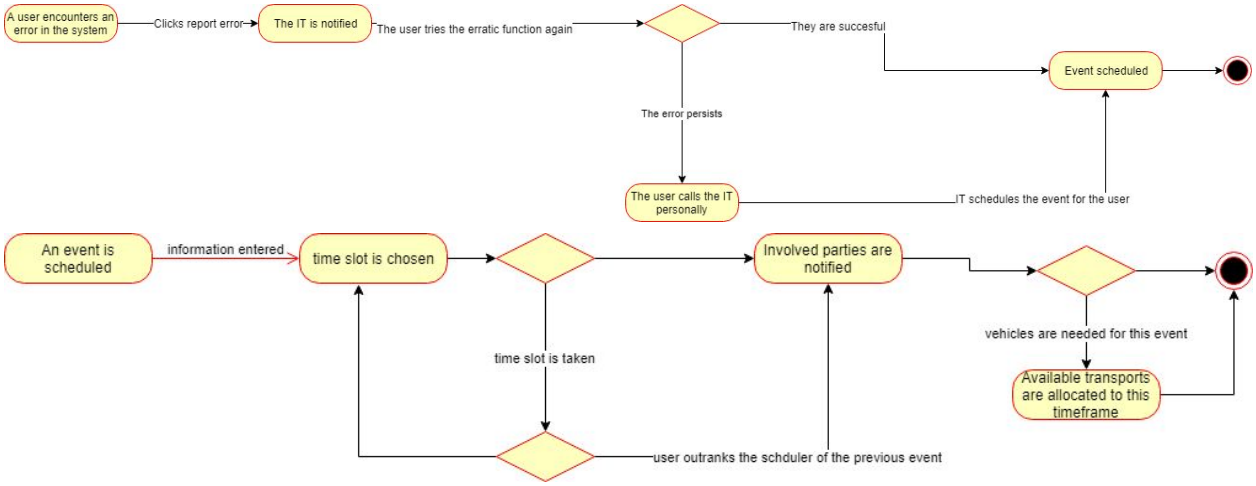


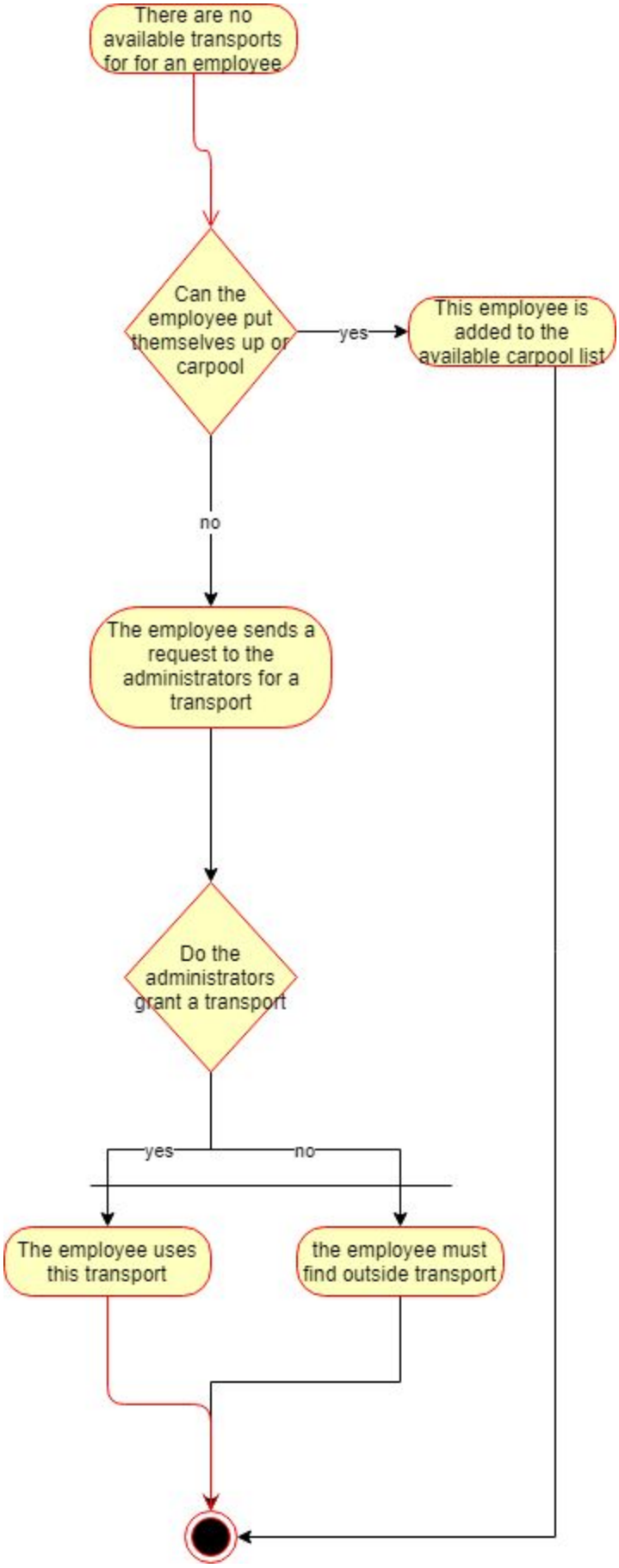


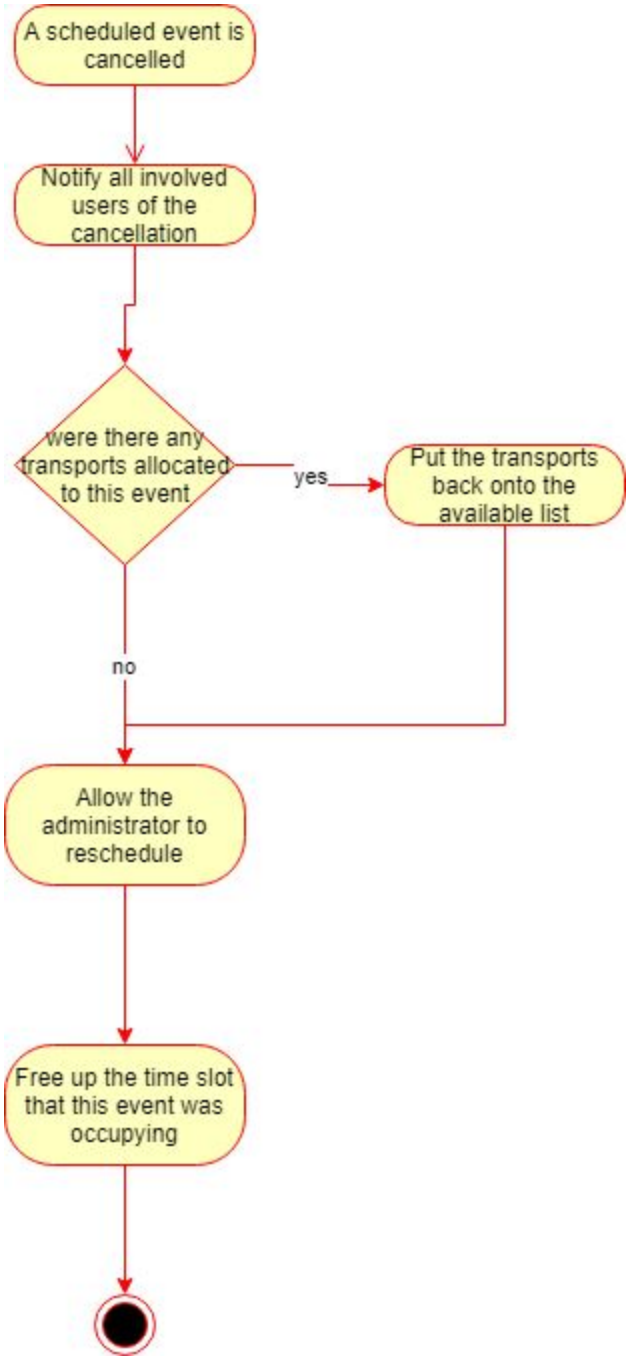
Explanation:
The portion that allows for IT employees (who maintain the system daily) to view transport demands is the System Metrics page. Usage statistics is a category that shows the frequency of what portions of the system are used the most as well as what types of transport were used the most on certain days. This information aligns with what was described in the requirements document and it should be noted that company executives who dictate financial decisions would have access to this portion of the system. IT employees also have the ability to view the different types of errors reported as well as the exact details typed out within the reports - as outlined in the requirements document UML diagram and scenarios - which allows them to make adjustments to the system to make it more efficient.

UML Diagrams




A user encounters an error in the system
Clicks report error
The IT is notified
The user tries the erratic function again
They are succesful
Event scheduled
The error persists
The user calls the IT personally
IT schedules the event for the user
An event is scheduled
information entered
time slot is chosen
Involved parties are notified
vehicles are needed for this event
Available transports are allocated to this timeframe
time slot is taken
user outranks the schduler of the previous event


There are no available transports for for an employee
Can the employee put themselves up or carpool
yes
This employee is added to the available carpool list
no
The employee sends a request to the administrators for a transport
Do the administrators grant a transport
yes
no
The employee uses this transport
the employee must find outside transport


A scheduled event is cancelled
Notify all involved users of the cancellation
were there any transports allocated to this event
yes
Put the transports back onto the available list
no
Allow the administrator to reschedule
Free up the time slot that this event was occupying

Team Member Contributions

At the start of this project, we attempted to evenly split up duties amongst our team members in order to maximize the efficiency and speed of our work. Then, the plan was for each member to read over the other members' contributions to proofread for grammatical errors, revise it so that the report has a good and easy-to-read flow, and make sure we're all thinking about the system in a cohesive way.

In Part 1 the entire team brainstormed on classes and methods to include within the class diagram. The duties assigned afterward were as follows: Reagan was to create the class diagram using the team's outline and provide a high-level description of the functionality of each class within the design (parts #1 and #2), Jack was to develop the preliminary visuals and verify that they match the details outlined within the project's requirements documentation (part #4), and Jackson was to create five different UML diagrams (sequence, state, etcetera) to elaborate more on the functions of the system (part #5). With each portion of the project other members were to review and aid any team members who were struggling. All team members were then to come together and review the entire document.

These duties were fulfilled as dictated. Each team member completed their tasks. Jack typed up the team contributions section and aided Reagan in a few high level descriptions..

Regarding communication about two were had via text initiated by Jack. All members responded. Most communication was done through discord, 5 text conversation total - 4 being by Jack and 1 being by Reagan - all members responded. The team held two meetings via discord voice chat - once to plan out the project and divy out assignments, the other to collaborate on the project itself and ask questions for clarification. Communication became much more consistent in this portion of the project. All members started later on their portions of the project than expected due to unforseen circumstances.