

CS4400 Database Project

Spring Semester 2016

Version 1.0

GTTrain.com

Look at the last page for edits made for each version. Please read the entire description of the project before starting to work on it.

Purpose of the Project

Analyze, specify, design, implement, document and demonstrate an online system for buying train tickets at GTTrain.com. You are required to use the classical methodology for database development. The system should be implemented using a relational DBMS that supports standard SQL queries. Class administrators will provide you with information about how to access a college-managed MySQL server in order to implement your database and the application. The professors must approve any other alternative implementations. ***In no circumstances can you use a tool that automatically generates SQL or automatically maps programming objects into the database. You also cannot use any other software like Access.*** Ask professors or TAs if you have doubts in which tools/languages/software are allowed.

Project Phases

The three phases of the project cover the following work-processes from the Classical Methodology for Database Development (see notes on T-square under resources). Slides on database design methodology will be useful for phases I and II: These slides have been posted on T-square.

Groups

Project groups may have 3 or 4 members. Groups of more than 4 or less than 3 will **NOT** be allowed. You are allowed to form groups across the three sections (A, B & C) of the class. A group may remove a member from further participation in the group when Phase I is turned in or when Phase II is turned in. A written

notification with a proper justification must be provided to the professor and the head TA at that time in hard-copy form.

Deliverables

Phase I (Soft copy and hard copy)

Deadline: Feb 18

The deliverables include:

1. A cover page listing all members in the team with their respective sections, GT official email addresses and T-square usernames.
2. Enhanced Entity Relationship (EER) Diagram
3. Information Flow Diagram
4. A list of logical constraints that will be enforced. Do not include any constraints that can be shown in the EER diagram, but rather semantic, business logic related constraints. You are required to include at least three constraints, although a fully-specified system will probably have more than that. **Constraints that can be specified directly using ER notation will not count towards the three required.** Constraints related to data type are not accepted as constraints.
5. Any assumptions made with explanations.

Notes:

1. The EER must capture the constraints of the system as much as possible whenever applicable, i.e. total participation, super/sub class, weak entities.
2. The design of your system must satisfy all the constraints. You are allowed to make up additional assumptions and constraints as long as they do not conflict with the specified constraints and requirements. If possible, those additional assumptions and constraints should be included in the ER diagram. You must list all your assumptions and constraints; otherwise TA would mark your ER diagram wrong since they would not be able to know

you have made your own assumptions.

Each group needs to turn in one hard copy (only one for the entire group), and each group member should upload an electronic copy on T-Square individually. Group numbers will be assigned to the groups after they are declared in Phase 1. **Please write down your Group Number clearly on all subsequent submissions.**

Phase II (Soft copy and hard copy)

Deadline: Mar 17

The deliverables include:

1. A cover page listing all members in the team with their respective sections, GT official email addresses and T-square usernames, and group number assigned in phase 1.
2. Copy of the ER Diagram (either from phase I (with any revisions) or from the solution provided)
3. Copy of the Information Flow Diagram from phase I (either from phase I (with any revisions) or from the solution provided)
4. Relational Schema Diagram (Identify primary and foreign keys and show referential integrity using arrows)
5. Create Table statements, including domain constraints, integrity constraints, primary keys, and foreign keys.

Note:

1. **Only one hard copy** should be turned in for the entire group, and each group member should upload an electronic copy on T-Square individually.

Phase III (Soft copy and hard copy)

Electronic Submission Deadline: Apr 24

Project Demo Dates: Apr 25, 26

The electronic deliverables include:

1. A cover page with the group number and the group members' names.
2. A text file with all SQL statements for each task (follow the template in the phase II design methodology)

Note: A set of SQL statements may be required in order to complete one task. However, in such cases, the last SQL statement should show the output according to the specification. Views and nested queries may be used to support the tasks.

3. For heavy weight option, you also need to submit your source code. You need to develop the entire application as a stand-alone application including the front end, menu options and the control flow. The application must have all functionalities described in this document.

Note: Prior to the demo, the TAs will give guidelines for populating the database with data. The database has to be populated with this data set prior to the demo. 5% will be deducted from the grade otherwise.

You need to submit your electronic copy of phase 3 to T-Square before Apr 24.

On demo day:

Bring your laptop and make sure you have a text file (soft copy) with all your SQL queries just in case that your application does not work. More details about demo will be discussed later this semester.

Grading

The project will consist of three phases (deliverables) as well as a final demo to the TA.

Phase I and Phase II of the project are each worth 10% credit.

Phase III (20% for heavy-weight or 10% credit for light-weight, depending on option):

Heavy Weight Option (20 %): The students would be required to use the embedded SQL feature of MySQL which allows you to embed SQL statements in a standalone application.

Light Weight option (10%): The students would be required to demo the SQL queries on the MySQL console. Those who choose the light weight option would be required to take the Final exam.

Note that you can always change your option until the demo starts. Once TA starts to demo your project, you cannot change heavy-weight option to light-weight or vice versa.

Final Exam (10%): This would be only taken by students who have opted for the lightweight phase III. Under no circumstances would a heavy weight option student be allowed to take the Final.

GTTrain.com is an online train ticket reservation website. There are two types of users using this website, customers and managers. The customers can make reservation, update departure date, cancel a reservation and view/provide comments about their trip experiences through this website. The managers can only use this website to view different kinds of reports.

The following sections contain a functional description of the system along with some mockup screens. Each section would explain a particular functionality and then present an example screen about it. **You don't have to follow the UI designs.** These mockups are just for helping you to understand all the functionalities. A complete reorganization of the user interface is permissible as long as your application supports all the functionality listed below. The sections have been grouped by customer's functionalities and managers' functionalities.

For heavy option, you may implement the project as a traditional standalone application (e.g., using Java GUIs) or as a web application (e.g., using a web scripting language like PHP). There is no restriction on the choice of language (e.g., Java, Python, Javascript). We will also send an announcement about which languages/tools/software/platforms are allowed later this semester. (Do ask the professors for permission if in doubt.)

1. Log In

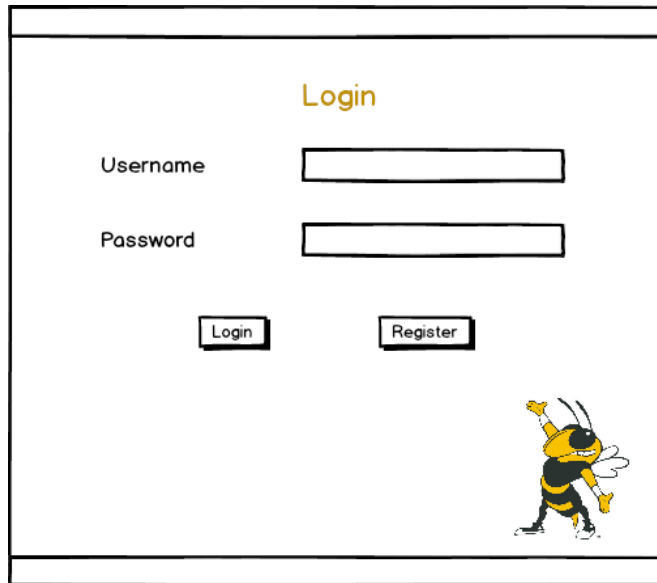
A mockup of a login screen. At the top center, the word "Login" is written in a yellow, sans-serif font. Below it, there are two input fields: the first is labeled "Username" and the second is labeled "Password". Both labels are in a black, sans-serif font. Below the input fields, there are two buttons: "Login" and "Register". Both buttons are rectangular with a black border and a light gray background. In the bottom right corner of the screen, there is a cartoon bee character with a yellow and black striped body, wearing white gloves and shoes, and having its arms raised in a celebratory gesture.

Fig 1: Log in

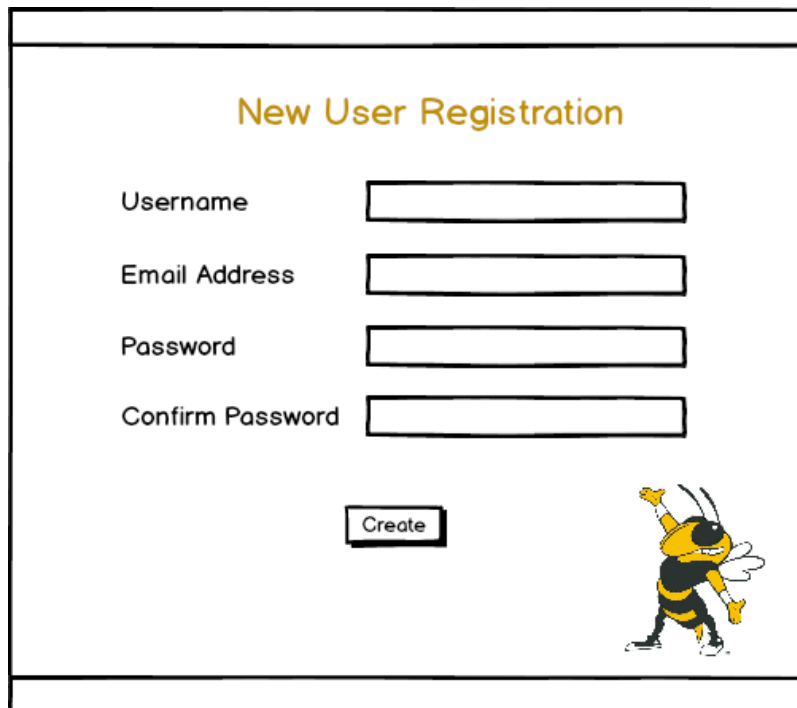
Fig 1 shows the login screen. All users must login before using this application. A valid username and password combination is required. If the user provides invalid login credentials, an error message should be shown on the screen.

If a customer does not have an account yet, s/he can click on the register button to create an account.

Note:

1. Username is **unique** for every user.
2. Since the customers and the managers use the same login screen, you need to check if the user is a customer or a manager.

2. New User Registration



The image shows a web form titled "New User Registration" in orange text. Below the title are four input fields: "Username", "Email Address", "Password", and "Confirm Password". Each field is represented by a rectangular box. Below the "Confirm Password" field is a "Create" button. To the right of the button is a cartoon bee character with yellow and black stripes, wings, and a smiling face.

Fig 2: New User Registration

After clicking register button in Figure 1, the user will be directed to the new user registration page.

The system needs to verify that all fields are filled, the username is available, password and confirm password are the same, and the email address is valid and **unique**. An error message should be shown if any of these requirements fails.

Notes:

1. This screen is **only** for creating customer accounts. We assume all managers' accounts have already been created behind the scene.
2. Each customer account has a unique email address associated with it while manager account does not.

Customer Functionalities

1. Choose functionality (customer view)

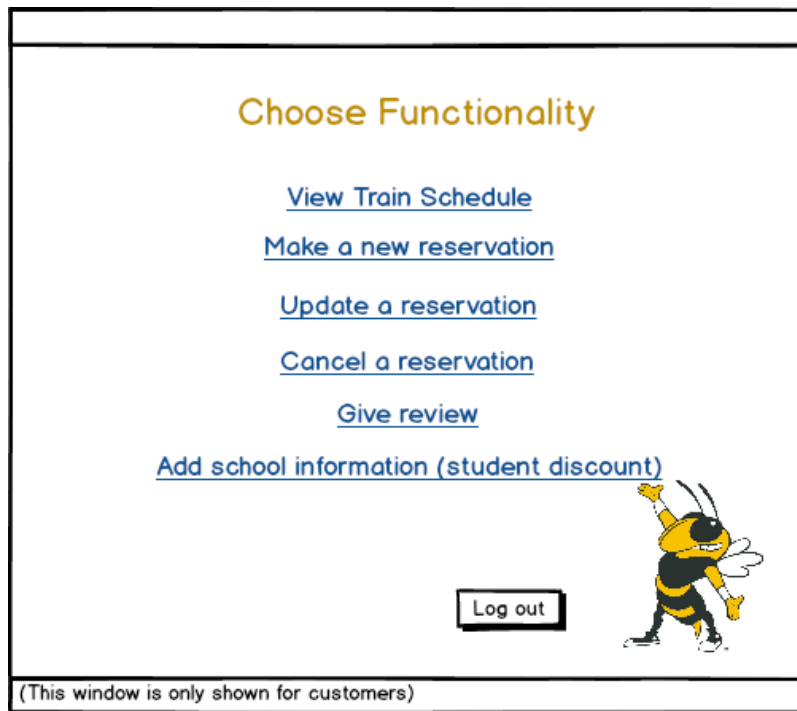
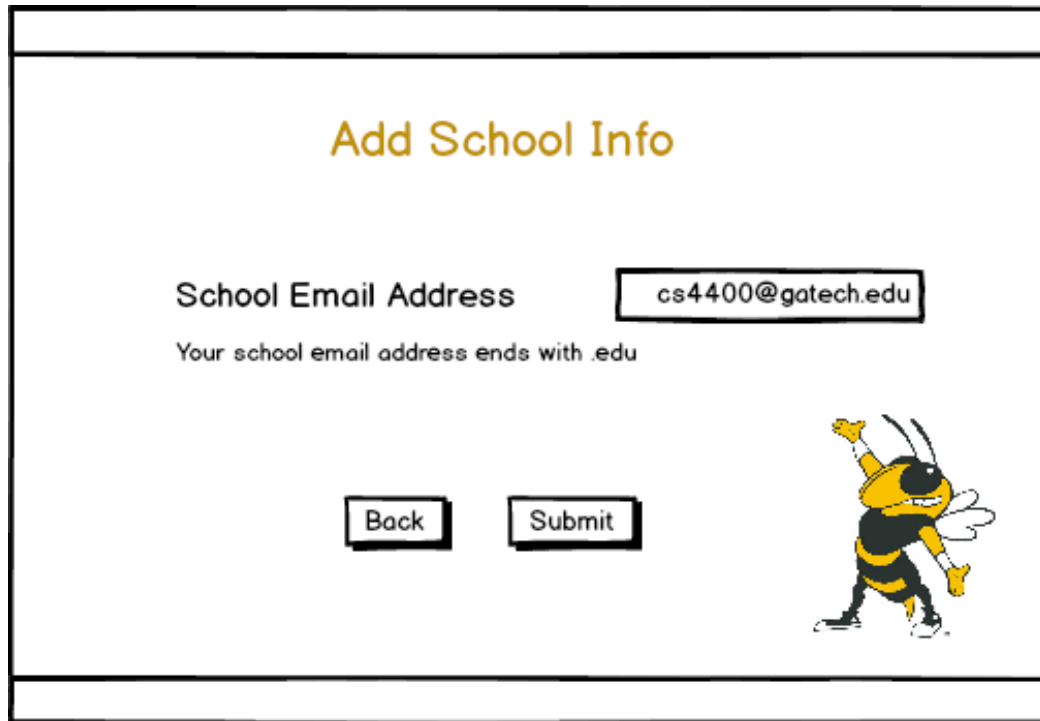


Figure 3: Choose Functionality (customer view)

After logged in as a customer, s/he would be taken to this window which lists out all functionalities for a customer. S/he can click on a functionality and be directed to the corresponding screen.

2. Add School Info



The screenshot shows a web form titled "Add School Info" in orange text. Below the title, there is a label "School Email Address" and a text input field containing "cs4400@gatech.edu". Underneath the input field, a hint text reads "Your school email address ends with .edu". At the bottom left, there are two buttons: "Back" and "Submit". At the bottom right, there is a cartoon bee character with its arms raised in a celebratory gesture.

Figure 4: Add School Info

There is a discount for students (20% off). To take the discount, all customer needs to do is to enter his/her official school email address (the one that ends with .edu).

Notes:

1. You do not need to store the school email address, but you do need to know whether the customer is a student or not. (You need this information when calculating the total cost of the reservation.)
2. You do not need to implement the actual email verification. We assume the customer will be able to take student discount as long as the email address ends with .edu. (In other words, you do not need to check if this email address is valid/unique or actually belongs to the customer.)


3. View Train Schedule

View Train Schedule

Train Number

2163 Express

Search



View Train Schedule

Train (Train Number)	Arrival Time	Departure Time	Station
2163 Express		11:25am	Boston(BBY)
	11:50am	11:55am	New London
	12:35pm	12:40pm	Windsor
	1:53pm	1:57pm	Stamford
	2:45pm	2:55pm	New York(Penn)
	3:50pm		Washington DC(Union)

Back




Figure 5: View Train Schedule

Each train route has a **unique** Train Number and a customer can view the train schedule using that number. The customer types in the train number and clicks on the search button, and then s/he should be taken to another screen which shows the schedule of the corresponding train. If the customer provides an invalid train number, an error message should be shown on the screen.

Take 2163 Express (as shown in Figure 4) as an example, it leaves Boston BBY at 11:25am, arrives at New London station at 11:50am and then departs from New London at 11:55am.

Note:

We assume the schedule of a train is same for everyday.

4. Make a reservation

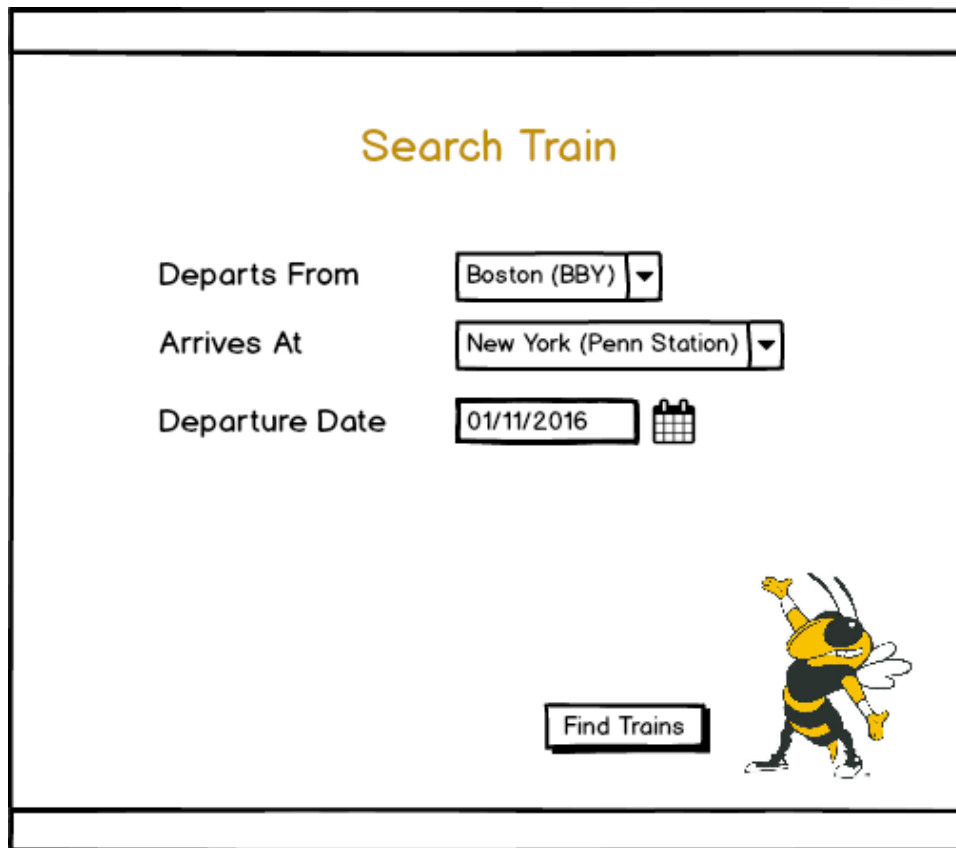
A screenshot of a web form titled "Search Train" in orange text. The form contains three input fields: "Departs From" with a dropdown menu showing "Boston (BBY)", "Arrives At" with a dropdown menu showing "New York (Penn Station)", and "Departure Date" with a text box showing "01/11/2016" and a calendar icon. Below these fields is a "Find Trains" button and a cartoon bee character.

Figure 6: Search Train

After choosing Make a Reservation on the Choose Functionality screen, the customer should be taken to this window, where s/he could search a train. Departure station and arrival station should be chosen from a dropdown menu.

Note: All train stations should be stored in the database. Each station has a **unique** combination of name (for example, Penn Station) and a location (New York).

Select Departure

Train (Train Number)	Time (Duration)	1st Class Price ⬆ ⬇ ⬆	2nd Class Price ⬆ ⬇ ⬆
2151 Express	5:10am-8:44am 3hr34min	<input type="radio"/> \$250	<input type="radio"/> \$135
95 Regional	6:15am-10:20am 4hr5min	<input type="radio"/> \$220	<input type="radio"/> \$115
2163 Express	11:25am-2:45pm 3hr30min	<input type="radio"/> \$220	<input checked="" type="radio"/> \$115
179 Regional	6:51pm-10:50pm 3hr59min	<input type="radio"/> \$150	<input type="radio"/> \$76
67 Regional	9:36pm-2:15am(Jan 12) 4hr39min	<input type="radio"/> \$100	<input type="radio"/> \$49

Back

Next




Figure 7: Select Departure

After clicking on Find Trains, a list of trains that fit the customer's search criteria should be displayed. Each train has a **unique** train number, schedule and prices for first class and second class.

In this example, the customer selects a second class ticket of 2163 Express, which departs from Boston at 11:25am and arrives at New York at 2:45pm.

Travel Extras & Passenger Info

Number of Baggage

3

▲▼

Every passenger can bring up to 4 baggage. 2 free of charge, 2 for \$30 per bag

Passenger Name

Alier Hu

Back

Next




Figure 8: Travel Extras & Passenger Info

After choosing a train, the customer should be directed to this window. Each passenger can bring up to four baggage, 2 free of charge and 2 for \$30 per bag. In the mockup, the customer indicates s/he would carry 3 baggage, so s/he needs to pay \$30 for the 3rd bag. Also, the customer needs to enter the passenger name.

Make Reservation

Currently Selected


Train (Train Number)	Time (Duration)	Departs From	Arrives At	Class	Price	#of Baggages	Passenger Name	Remove
2163 Express	Jan 11 11:25am-2:45pm 3hr30min	Boston(BBY)	New York(Penn)	2nd Class	\$115	3	Alier Hu	<input type="button" value="Remove"/>

Student Discount Applied.

Total Cost 116

Use Card 2541 [Add Card](#)

[Continue adding a train](#)



$(115+30)*0.8=116$

Figure 8: Make Reservation 1

After entering all information needed, the customer should be taken to this window. Since the customer has entered his/her school information in Figure 4, s/he can get the student discount. The original total cost was \$115(train ticket price) + \$30 (3rd baggage) = \$145. After applying the student discount, the total cost becomes $\$145 * 0.8 = \116 .

The customer can also delete the selected train ticket by clicking the Remove button. The total cost should be updated accordingly.

At this stage, the customer can either choose a payment method and make a reservation, or continue adding a train ticket to the reservation. Let's say the customer clicks Continue Adding a Train and goes through Figure 6-8 again. The final Make Reservation window should look like below:

Make Reservation

Currently Selected


Train (Train Number)	Time (Duration)	Departs From	Arrives At	Class	Price	#of Baggages	Passenger Name	Remove
2163 Express	Jan 11 11:25am-2:45pm 3hr30min	Boston(BBY)	New York(Penn)	2nd Class	\$115	3	Alier Hu	<input type="button" value="Remove"/>
2543 Regional	Jan 20 2:30pm-4:50pm 2hr20min	New York(Penn)	Washington DC(Union)	1st Class	\$100	2	Alier Hu	<input type="button" value="Remove"/>

Student Discount Applied.

Total Cost 196

Use Card 2541 [Add Card](#)

[Continue adding a train](#)



$(115+30+100)*0.8=196$

Figure 9: Make Reservation 2

The window shows the customer has added another train ticket (New York - Washington DC, 1st class, 2 baggage) to his/her reservation. The total cost should be updated accordingly. (\$115 for 2163 Express+ \$30 for 3rd baggage+ \$100 for 2543 Express = \$245. Then apply student discount: \$245*0.8=\$196)

Before the user can submit his/her reservation, s/he needs to have at least one card stored in the system that can be used to pay for his/her reservation. The customer can manage all his/her cards in the Payment Information window.


Payment Information

Add Card

Name on Card

Card Number

CVV

Expiration Date 

Delete Card

Card Number ▼




Figure 10: Payment Information

After clicking Add Card button (in Figure 9), the customer should be directed to the Payment Information page, where the customer can add/ delete a card. All fields need to be filled and the expiration date should be greater than today.

Note: The customer can use **only one** card for transaction of one reservation, and s/he cannot delete a card if this card is being used in a transaction which has not ended yet. (the departure date has not passed)

After clicking Submit, the customer should be taken back to Figure 9, where s/he can choose a card to pay for the reservation from a dropdown menu.

5. Confirmation Screen

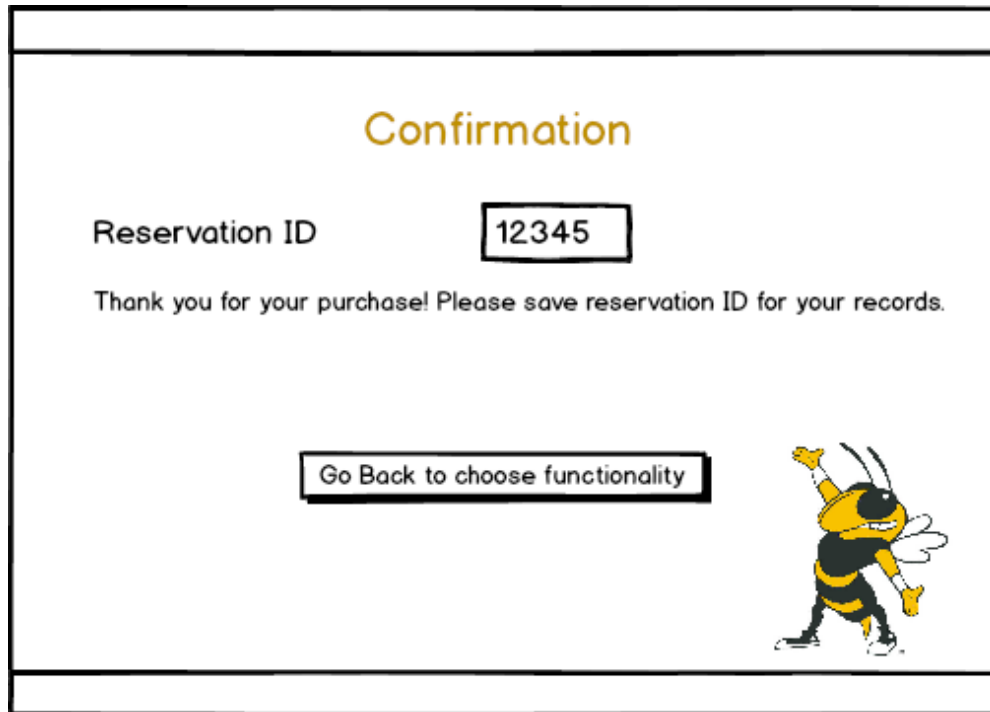


Figure 11: Confirmation Screen

After a reservation has been made, a reservation ID is generated for the customer. The confirmation screen should show the reservation ID.

6. Update Reservation

Unlike real world system, GTTrain.com only allows customers to update the departure date.

Update Reservation

Reservation ID

12345

Search

Back




Figure 12: Update Reservation 1

An error message should show if the reservation ID cannot be found or the reservation was not made by the customer.

Update Reservation

Select	Train (Train Number)	Time (Duration)	Departs From	Arrives At	Class	Price	#of Baggages	Passenger Name
<input checked="" type="radio"/>	2163 Express	Jan 11 11:25am-2:45pm 3hr30min	Boston(BBY)	New York(Penn)	2nd Class	\$115	3	Alier Hu
<input type="radio"/>	2543 Regional	Jan 20 2:30pm-4:50pm 2hr20min	New York(Penn)	Washington DC(Union)	1st Class	\$100	2	Alier Hu

Back

Next




Figure 13: Update Reservation 2

The customer should select which train ticket s/he would like to update. S/he can only update 1 ticket at a time. If s/he needs to change multiple train tickets, s/he should go through this process multiple times.

Update Reservation

Current Train Ticket

Train (Train Number)	Time (Duration)	Departs From	Arrives At	Class	Price	#of Baggages	Passenger Name
2163 Express	Jan 11 11:25am-2:45pm 3hr30min	Boston(BBY)	New York(Penn)	2nd Class	\$115	3	Alier Hu

New Departure Date

Updated Train Ticket

Train (Train Number)	Time (Duration)	Departs From	Arrives At	Class	Price	#of Baggages	Passenger Name
2163 Express	Jan 13 11:25am-2:45pm 3hr30min	Boston(BBY)	New York(Penn)	2nd Class	\$115	3	Alier Hu

Change Fee

Updated Total Cost

Figure 14: Update Reservation 3

A \$50 change fee would occur if a customer updates a ticket, and the total cost of the whole reservation should be updated accordingly. (The original total cost was \$196, plus the \$50 change fee, so the update total cost is \$246)


Note:

1. **Only** the departure date can be changed.
 2. An update should be made at least 1 day earlier than the departure date.
- After that, the customer cannot cancel or update a reservation.

7. Cancel Reservation

Cancel Reservation

Reservation ID



Cancel Reservation

Train (Train Number)	Time (Duration)	Departs From	Arrives At	Class	Price	#of Baggages	Passenger Name
2163 Express	Jan 13 11:25am-2:45pm 3hr30min	Boston(BBY)	New York(Penn)	2nd Class	\$115	3	Alier Hu
2543 Regional	Jan 20 2:30pm-4:50pm 2hr20min	New York(Penn)	Washington DC(Union)	1st Class	\$100	2	Alier Hu

Total Cost of Reservation

Date of Cancellation

Amount to be Refunded




Figure 15: Cancel Reservation

A customer can cancel a reservation and the amount to be refunded is calculated as follows:

1. 80% of the original total cost will be refunded if the customer cancelled the reservation more than 7 days earlier than the earliest departure date. (In the mockup, there are two departure dates: Jan 13 and Jan 20. The earliest departure date is Jan 13)

2. 50% of the original total cost will be refunded if the customer cancelled the reservation more than 1 day but less than 7 days earlier than the earliest departure date.

3. A \$50 cancellation fee will be deducted from the refund.

For example: (See mockup)

The earliest departure date of this reservation is Jan 13, and the original total cost is \$246.

- If a customer cancels before Jan 6: 80% of \$246 should be refunded, and then after deducting the \$50 cancellation fee, the refund should be \$ 146.8. ($246 \times 0.8 - 50 = 146.8$)
- If a customer cancels any time between Jan 7- 12: 50% of \$246 should be refunded, and then after deducting the \$50 cancellation fee, the refund should be \$73. ($246 \times 0.5 - 50 = 73$)
- No cancellation can be made after Jan 12.

Note:

1. A cancelled reservation should not be removed from the database, but the system needs to know if a reservation has been cancelled or not.

2. Cancellations should be made at least 1 day before the earliest departure date of the reservation.

3. The refund should **not** be negative. (For example, if the original total cost was \$20, so then after deducting the cancellation fee, the refund will be negative. In that case, the refund would just be 0.)

4. Remember to recalculate and update the total cost of the reservation. Take the example from the mockup, the total cost of the reservation would become $\$246 - \$146.8 = \$99.2$.

5. Remember that you **cannot** cancel or update a cancelled reservation. An error message should be displayed if a customer attempts to do so.

[8. View Review](#)

A customer can view reviews of a specific train.


View Review

Train Number

2163 Express

Back

Next



View Review

Rating	Comment
Good	The train was on time.
Neutral	
Bad	The bathroom was very dirty.

Back to Choose Functionality




Figure 16: View Review

9. Give Review

A customer can also provide review. We assume that all customers will only provide reviews after their trips, so you do not need to check it.

Give Review

Train Number

2163 Express

Rating

Good ▼

Comment

Good

Submit




Figure 17: Give Review

The customer can select a rating from a range of values: very good, good, neutral, bad, very bad.

Note:

1. Customers can leave the comment part blank, but the train number and rating fields must be filled.
2. If the train number is unavailable, an error message should be displayed.

Manager Functionalities

1. Choose Functionality (manager view)

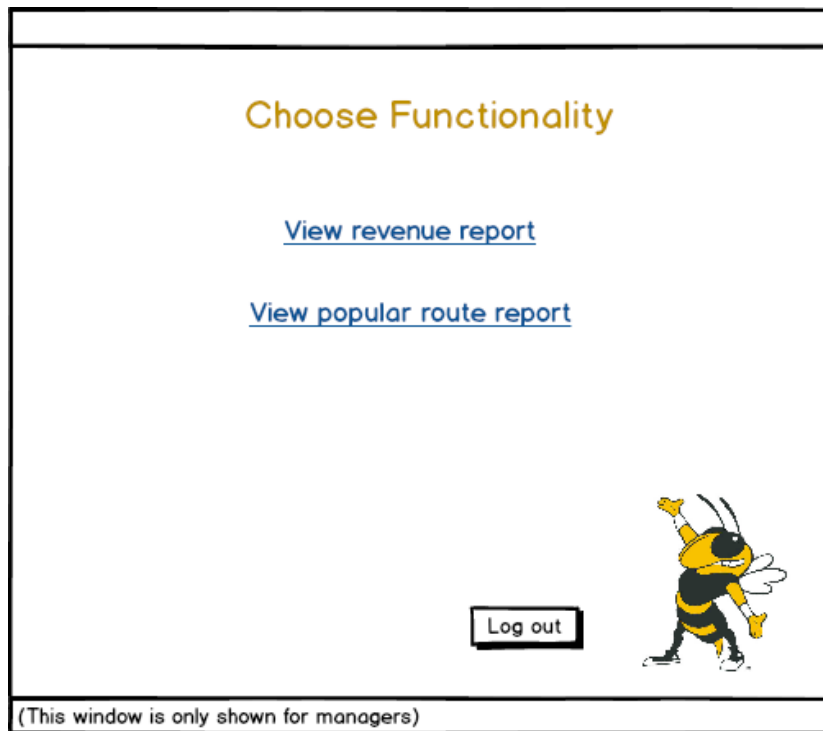


Figure 18: Choose Functionality (manager view)

If the user has logged in as a manager, s/he should be taken to this window. As a manager, the user can only view revenue report and view popular train report.

Hint:

1. You need to consider whether these reports should be stored in the database.

2. View Revenue Report

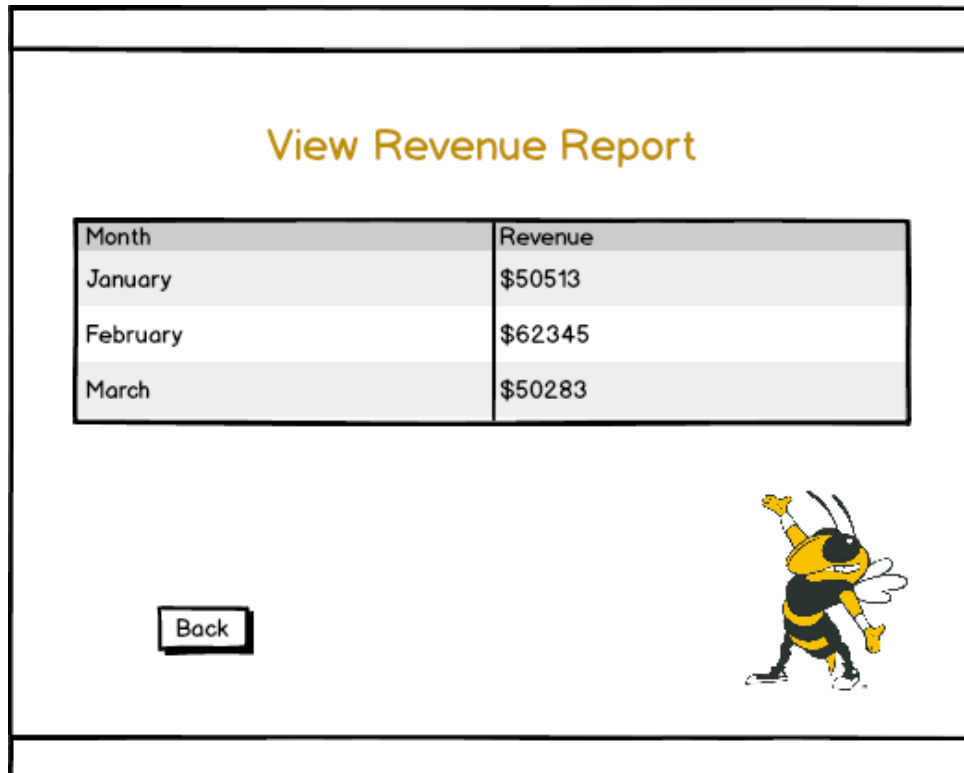


Figure 19: View Revenue Report

This report shows the total revenue of each month.

Note:

1. For this report, you only need to show three months.
2. Do not hardcode the reports. You should do this using SQL statements.

3. View Popular Train Report

Figure 20: View Popular Train Report

View Popular Route Report

Month	Train number	#of Reservations
January	2451 Regional	613
	2708 Regional	413
	924 Express	390
February	5143 Regional	728
	2708 Regional	634
	4022 Regional	408
March	5143 Regional	597
	4022 Regional	516
	924 Express	451

[Back](#)



This report shows top 3 most popular train (calculated by number of reservations) of each month.

Note:

1. Use the departure date for calculating the month
2. Similarly to the previous report, you only need to show three months.
3. Do not include cancelled reservations.