

Indermohan Gill 216542045

EECS 3311 Project Report

I could not get my zip file to store the jars I used inside of it so they must be downloaded from my Google Drive.

To start and run the program, first download the project zip file and import it in eclipse workplace. Then go to the “Configure Build Path” screen and remove the external jars that are giving errors. Next go to <https://drive.google.com/file/d/1NMemZ5MGagdKzcSGdWzH9raNRVrsLZdE/view?usp=sharing> and download the zip containing the jar files I used. Extract the jar files into a folder and return to eclipse. Go to “Configure Build Path” then from the top select “Libraries”, then “Add External Jars”. Now select the 3 jars that were in the zip file and apply. Now the project should have no errors.

NOTE: there is an account already created for an admin to use, its username is **admin1** and its password is **admin123**

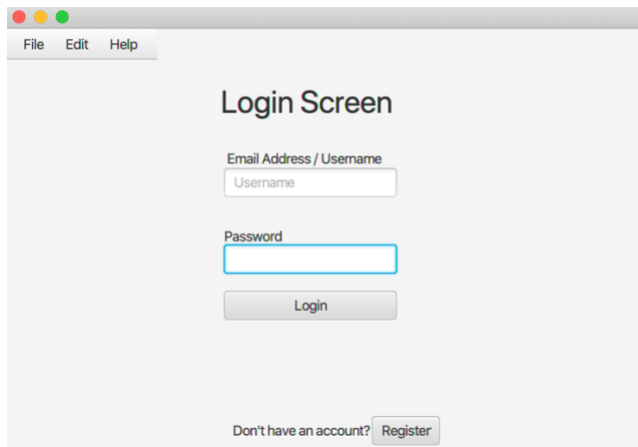
NOTE: For the CSV files, the locations must be changed in classes:

AdminPaymentStatus.java, Booking.java, Customer.java, NewEnforcerController.java,
OfficerManageParkingController.java, OfficerManageRequest.java,
OfficerViewBookingsController.java, ParkingEnforcer.java, RegisterController.java,
RemoveEnforcerController.java and SystemAdministrator.java

Assumption: Customer first books a parking, enforcer grants it, customer pays for it, admin changes payment status.

Requirement 1: User Login

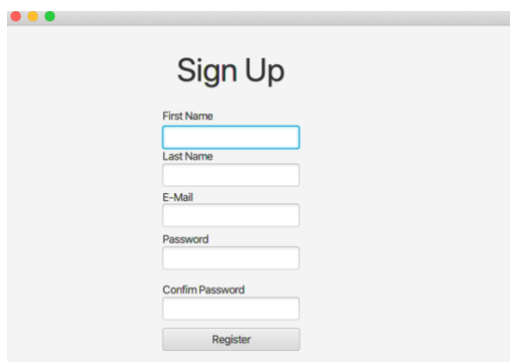
Once the program is run, this window will be shown to the user. Depending on whether the user is already in the system database, they can login. Once the fields have been filled, clicking the “Login” button shall log you in as long as the user and the password are correct and in the database.



A screenshot of a macOS-style window titled "Login Screen". The window has a menu bar with "File", "Edit", and "Help". The main content area contains a form with the following elements: a label "Email Address / Username" above a text input field containing the placeholder "Username"; a label "Password" above a text input field; a "Login" button below the password field; and at the bottom, the text "Don't have an account?" followed by a "Register" button.

Requirement 2: Customer Registration

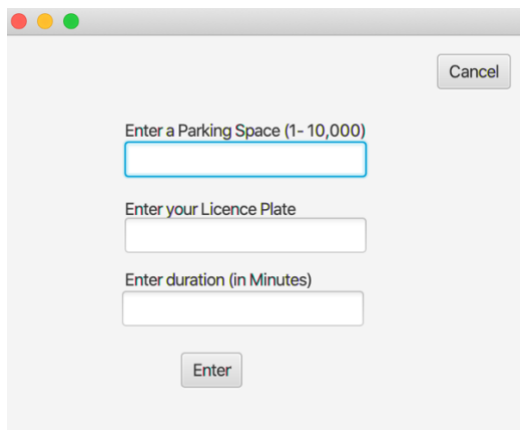
In the Login Screen, there is a button at the bottom that says “Register”. Clicking this button will open a new window with fields for the customer to fill out to create their account. It checks if the two passwords entered are the same and stores all the data in the system database.



A screenshot of a macOS-style window titled "Sign Up". The window has a menu bar with three colored window control buttons (red, yellow, green). The main content area contains a form with the following elements: a label "First Name" above a text input field; a label "Last Name" above a text input field; a label "E-Mail" above a text input field; a label "Password" above a text input field; a label "Confirm Password" above a text input field; and a "Register" button at the bottom.

Requirement 3: Book a Parking Space

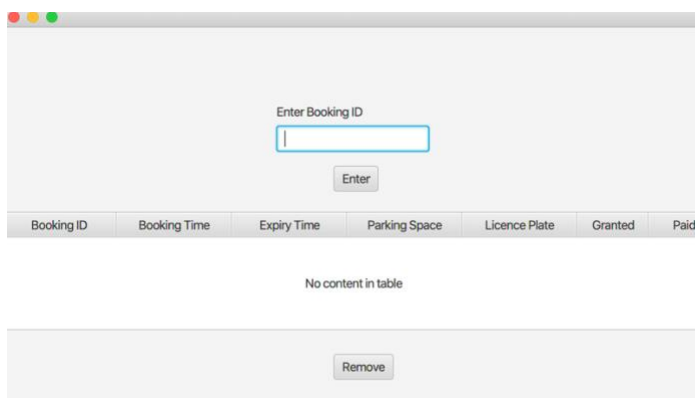
After a customer logs in, their screen looks as it does below. To book a parking space, the customer must click on the “Book Space” button and a new window will appear. This window asks the customer the location of the booking, expected booking time, duration of the booking and customer license plate. After all fields have been filled with expected inputs, the window closes, and the system creates a new booking with that information.



A screenshot of a web application window titled "Book a Parking Space". The window has a light gray background and a standard macOS-style title bar with red, yellow, and green window control buttons. In the top right corner, there is a "Cancel" button. The main content area contains three text input fields, each with a label above it: "Enter a Parking Space (1- 10,000)", "Enter your Licence Plate", and "Enter duration (in Minutes)". Below these fields is an "Enter" button.

Requirement 4: Cancel a Booking

After the customer logs in, to cancel a booking, the customer must click on the button that says “Cancel Booking”. This will open a new window and requires the booking ID to be inputted into the textbox.

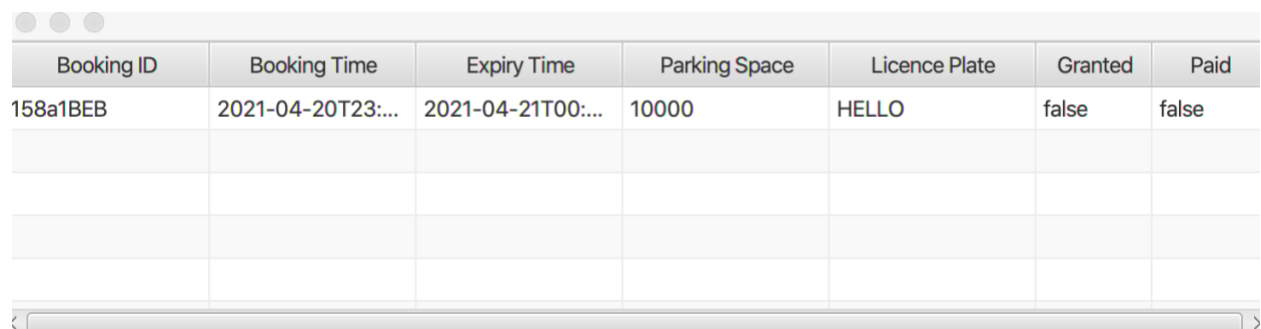


A screenshot of a web application window titled "Cancel Booking". The window has a light gray background and a standard macOS-style title bar with red, yellow, and green window control buttons. In the center, there is a text input field with the label "Enter Booking ID" above it. Below the input field is an "Enter" button. At the bottom of the window, there is a table with the following headers: "Booking ID", "Booking Time", "Expiry Time", "Parking Space", "Licence Plate", "Granted", and "Paid". Below the table headers, the text "No content in table" is displayed. At the very bottom of the window, there is a "Remove" button.

After the booking ID is entered, the customer must click “Enter” for the system to handle the input, the booking then shows up in the table below the button and its information is shown as well. Clicking the “Remove” button will cancel the booking so long as the expiry time of the booking has not passed.

Requirement 5.1: View Current Bookings (Customer)

After logging in, the customer must select “View Bookings” and will be directed to a new window.

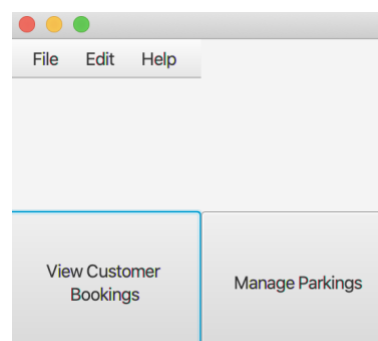


Booking ID	Booking Time	Expiry Time	Parking Space	Licence Plate	Granted	Paid
158a1BEB	2021-04-20T23:...	2021-04-21T00:...	10000	HELLO	false	false

Above is an example of a booking that has been added but not been granted by the Parking Enforcers yet, so it has not been paid for. The booking time and expiry time as of now are placeholders as the parking space has not been paid for and would begin at moment of payment.

Requirement 5.2: View Customer Booking Details (Parking Enforcer)

When a parking enforcer logs into the system, they are met with this window below.



After clicking on the “View Customer Bookings”, a new window is opened. To View a customer’s bookings, the parking enforcer must enter the customer’s First Name, Last Name and E-Mail.

Then, the requested and granted parking all show up.

A screenshot of a web application interface. At the top, there is a 'Cancel' button. Below it are three input fields: 'Customer First Name', 'Customer Last Name', and 'Customer E-Mail'. A 'Find Bookings' button is positioned below the input fields. Below the button is a table with the following headers: 'Booking ID', 'Booking Time', 'Expiry Time', 'Parking Space', 'Licence Plate', 'Granted', and 'Paid'. The table is currently empty, with the text 'No content in table' centered below the headers.

Requirement 6: Manage Parking Officer (Add and Remove)

A screenshot of a web application interface. At the top, there is a menu bar with 'File', 'Edit', and 'Help' options. Below the menu bar is a large empty area. At the bottom, there are three buttons: 'Add Parking Enforcer', 'Remove Parking Enforcer', and 'Change Payment Status of a Parking Space'.

This is the screen admins can view after logging in. If the admin chooses to add a new parking enforcer, after clicking the button, a new window opens that takes the input from admin.

A screenshot of a web application window with a light gray background. At the top, there are three colored window control buttons (red, yellow, green). Below them, the form contains five input fields stacked vertically, each with a label to its left: 'First Name', 'Last Name', 'E-Mail', 'Password', and 'Confirm Password'. At the bottom of the form is a single button labeled 'Add'.

After clicking “add”, the system will add the new parking enforcer and closes this window.

To remove a parking enforcer, after the button is clicked the window below is the one that opens.

A screenshot of a web application window for removing a parking enforcer. It features a light gray background and three window control buttons at the top. The main area contains a text prompt 'Enter the E-Mail of the Enforcer you wish to remove' above a single input field. Below the input field is an 'Enter' button. Underneath this is a table with three columns labeled 'First Name', 'Last Name', and 'Email'. The table is currently empty, with the text 'No content in table' centered below it. At the bottom of the window is a 'Remove' button.

It asks the admin to input the email of the enforcer it wishes to remove, then once the system finds the enforcer, it shows their details in the table to help visualize and confirm that the enforcer is the individual to be removed. After clicking the “Remove” button, the enforcer is removed from the system.

Requirement 7: Customer Payment

Cancel

Enter the parking space number to pay for

Enter

Pay For Parkings

Then once you have added the amount of parking to pay for, you are directed to this window.

Cancel

For PayPal click here

Payment Screen

Cardholder Name

Card Number

(MM / YYYY)

CVC

Confirm Payment

Requirement 8: Manage Parking Spaces

After the Enforcer clicks “Manage Parking Spaces” this window opens.

Cancel

Added Parking Spaces
30

Enter the number of the parking space you wish to add, remove or manage

Add Parking Space

Manage Booking Requests

Remove Parking Spaces

To check the booking requests for a particular space, it must be typed in the textfield then click “Manage Booking requests” to see this window.

A screenshot of a web application window. At the top left, there are three colored window control buttons (red, yellow, green). Below them is a 'Cancel' button. In the center, there are two buttons: 'Grant' and 'Decline'. The 'Decline' button is highlighted with a blue border. Below these buttons is a table with the following data:

Booking ID	Booking Time	Expiry Time	Parking Space	Licence Plate	Granted	Paid
cinXTxpV	+999999999-12...	+999999999-12...	300	HELLO	false	false

At the bottom of the window, there is a horizontal scrollbar.

Requirement 9: Manage Payment Status

After an admin clicks “Change Payment Status”, this window opens where the admin must enter the information of the customers’ parking that they wish to change the payment status of.

A screenshot of a web application window. At the top right, there is a 'Cancel' button. Below it, there are four text input fields with labels: 'Customer First Name', 'Customer Last Name', 'Customer E-Mail', and 'Parking Space Number'. At the bottom, there is a 'Change Payment Status' button.

Testing and Coverage

For the parking enforcer, I could not test the manage parking spaces as that was all done by the GUI.

For the customer, I could not test the payment or registration as that was all done by the GUI.

For system administrator I could not test the change payment status as that was all done by the GUI.

For the testing of the bookings, the code for writing to the .csv files and .txt files were commented out as they required input from the GUI.

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
3311Project	14.0 %	997	6,147	7,144
src	10.2 %	698	6,147	6,845
application	10.2 %	698	6,147	6,845
AdminController.java	0.0 %	0	126	126
AdminPaymentStatusController.java	0.0 %	0	755	755
Booking.java	19.3 %	49	205	254
BookingDataType.java	0.0 %	0	50	50
CancelBookingController.java	0.0 %	0	253	253
Customer.java	76.7 %	132	40	172
CustomerController.java	0.0 %	0	186	186
CustomerPriceController.java	0.0 %	0	220	220
CustomerViewBookingsController.java	0.0 %	0	214	214
EnforcerDataType.java	0.0 %	0	21	21
LoginController.java	0.0 %	0	234	234
Main.java	0.0 %	0	28	28
NewEnforcerController.java	0.0 %	0	210	210
OfficerController.java	0.0 %	0	65	65
OfficerManageParkingController.java	0.0 %	0	536	536
OfficerManageRequestController.java	0.0 %	0	1,569	1,569
OfficerViewBookingsController.java	0.0 %	0	345	345
ParkingBookingController.java	0.0 %	0	134	134
ParkingEnforcer.java	90.9 %	110	11	121
PaymentController.java	0.0 %	0	213	213
RandomString.java	0.0 %	0	33	33
RegisterController.java	0.0 %	0	355	355
RemoveEnforcerController.java	0.0 %	0	279	279
SystemAdministrator.java	86.2 %	407	65	472
test	100.0 %	299	0	299
(default package)	100.0 %	299	0	299

Remarks

My initial design as used in the midterm was to use the Builder pattern. However, in my final design, I opted to use the Composite Pattern and Factory Pattern partially.