

Flow

- Problem Statement, Purpose, Target Audience (1min)
- Overall Use Cases, System Architecture (2min)
- End-User Demonstration 1 (5min)
 - Register (Auto Login)
 - Create Bookings
 - View Bookings
 - Edit Bookings
 - Delete Bookings
 - Logout
- End-User Demonstration 2 (5min)
 - Login
 - View Shop
 - Add Product to Cart
 - Update Product Quantity in Cart
 - Delete Product from Cart
 - Checkout Cart
 - Logout
- Design Principles (2min)
- SWE Principles (1min)

Design Patterns

- Architecture → Principle of Least Knowledge
 - Separate front-end and back-end code
 - Communication via REST APIs
 - Abstracts away implementation on both ends, which allows for greater ease in adding new features
- UI → Facade
 - Access core functionalities from 1 page
 - Loose coupling
 - Only fetch necessary data
- Front-End → DRY (Don't Repeat Yourself)
 - Create reusable UI components in React
 - Call these UI components whenever required instead of rewriting the same code
- Back-End → MVC Pattern
 - Models created based on required objects and resources
 - Controllers manipulate these models, and updates the view, which in this case is the response for a REST API call
 - Minimize coupling on the backend and multiple developers can work on different features simultaneously
- Back-End → Single Responsibility
 - Separated into various controllers or endpoints according to specific functionalities
 - Easy to debug and test individually

SWE Principles

- Frequent Refactoring
 - Continually improve the cleanliness, readability and reusability of our code
- Version Tracking using GitHub
 - Very easy for multiple developers to implement different functionality
 - Ensures we can fall back to a working version if any breaking change occurs
- Pair Programming
 - Multiple developers working together on the same feature
 - Builds common ownership of code
 - Ensures easy sharing of knowledge to improve code implementation
- Bi-Weekly Standup
 - Ensures all of us are aware of potential issues we are facing
 - Everybody is aligned on what to focus on

Scripts

Barry

Hello, I'll be demonstrating the registration procedure for a new user. My name is Barry, and I'm interested to sign up with StudyBuddy.

Clicking register will bring me to the sign up form where I have to include my personal details.

First, I will input my preferred username, then my email. The next step would be to include my name and my mobile number. Finally, I'm going to input my password. There is another field to confirm my password. After all is done, I can proceed to click register.

- First scenario, last name will be blank
- Email will not have @gmail.com
- Mobile number will be 96117452
- Password will be 1234567890

We can see here that there are some issues with the registration form. Firstly, neither of the name fields should be left blank. The page will prompt the user to give an input. Country code is also required while keying in my mobile number. Here we can see that it is not sufficient to only key in my email's username. I have to key in my proper email address. Next, user's passwords cannot be entirely numeric, it must contain letters and numbers.

- Second scenario, password will be 123barry, but password confirmation will be 122barry

After editing the relevant fields, I accidentally made a typo when confirming my password. The page thus prompts me and tells me to check again.

Finally, I have successfully registered and I will automatically be brought to the user's home page. Allow me to demonstrate the booking of study area procedure.

Clicking on study areas on the navigation bar displays all the available study areas ready for booking. Besides their locations, the remaining vacancies for all study areas are shown, together with the time they are open daily. After scrolling through, I decide to book a slot at Jurong Spring Community Club Room 1.

Clicking on book will bring me to another page for me to pick my date and time. This input raises an error message which prompts users to only book in 1 hour intervals. Furthermore, we can only start from the start of the hour. Successful input will lead me to the page where all my bookings will be displayed. This also shows that my booking has been confirmed.

- Start timing will be 14:34
- End timing will be 15:40

Upon further consideration, I realised that I am supposed to go for a health check up in the morning. Thus, I have to change my time. Clicking on edit will bring me to another page where I can shift my timeslot. I then change the time to later in the day and this will be updated under my bookings accordingly. It will also be displayed under upcoming bookings.

But on further thought, I think that I'll be too lethargic after my medical check up in the morning. Hence I don't think I want to study on that day anymore. I can easily proceed to delete my booking with just the click of a button.

Here I've demonstrated all the features of user bookings. I will proceed to log out from the portal and it will bring me back to the landing page.

Gary

Hi, I'm Gary, I am going to show you how to log in after your account has been registered. To login, I will need to click on Log in to enter the login page. If I do not have an account, I can simply click on the register button to register an account like Barry has shown just now. However, in my case, I already have an account so I will need to input a valid username and password. If I input either the wrong username or password, I will not be able to login. If the username and password is correct, I click on login. Then the system will log me in to the homepage for logged in users.

Login:

1. Click on login in navbar
2. Type in username and password (garyy)(test1234) (error login)
3. Delete y
4. Click log in

Next, I will demonstrate how the shop feature will work for StudyBuddy, to access the shop feature, I can click on Browse products to go to the shop, or (clicks to home), click on Shop at the navbar, both buttons bring me to the shop page. I want to buy 2 pens, a ruler and a textbook. To do that, I can click on the pen twice, the ruler once, and one textbook. Next, to verify that I have added them to my cart, I will click on Cart at the navbar. The items will show up in the cart.

Shop: (add to cart)

1. Click on Browse products
2. Go to home (top left)
3. click on shop on the navbar (demonstrate that both will bring you to the shop)
4. Click on 2 pens, a ruler and a textbook
5. Click on Cart at navbar

The current total price is \$21. After reviewing the cart, I realised that I actually wanted 2 rulers instead of 2 pens. I will need to reduce the amount of pens to 1, and add an additional ruler to cart, to do that, there are buttons here to edit the quantity of items i want. I will click on the minus button for the pen, and add button for the ruler. Then, I realised that i made another mistake, i wanted the physics textbook instead of math. What I can do is to delete the math textbook with this button. Then, I will click on Keep Shopping to go back to the store and add a physics

textbook. The cart summary will reflect how much the total price is according to my edits.

Edit in cart

1. Show total price
2. Remove a pen and add a ruler
3. Delete the math textbook
4. Go back to shop in the navbar
5. Add a physics book to cart
6. Go back to cart
7. Show cart and total price

Lastly, to check out my items, if I don't write the address and postal code, it will give me an error. If I leave either address or postal code empty, it will show me the error accordingly. After I type in a valid address and postal code, I will be able to check out. If it is successful, the site will bring me to the orders page, where I can see my previous orders. It will show the status of the order, the address to be delivered and when the order was made. That will be it for the shop features for StudyBuddy.

Checkout:

1. Click on checkout (show error for both empty)
2. Type (NTU) into address
3. Click on checkout (show error for postal code)
4. Delete address and type (639798) into postal code
5. Click on checkout (show error for address)
6. Type in Address and Postal Code (NTU, 639798)
7. Click on checkout