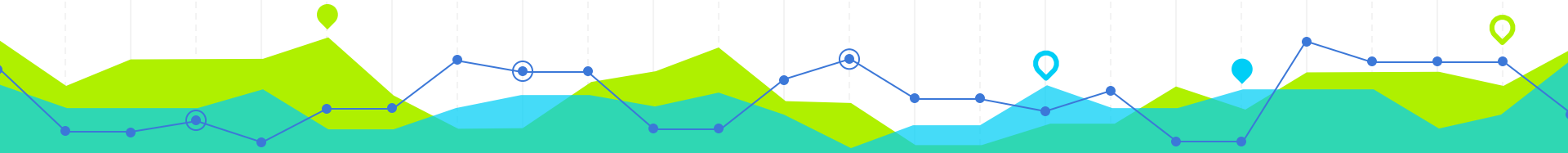
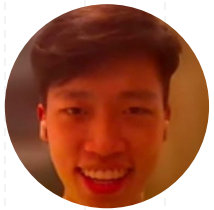


SWE Presentation

Team Fantastic 4

THE TEAM

Ang Kai Jun



Ivan Loke Zhi Hao

Chay Hui Xiang

Liu Liwen

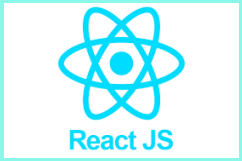
Ng Li Lin Evonne

Chang Dao Zheng

TwoTyred Application

TECH STACK

FRONTEND

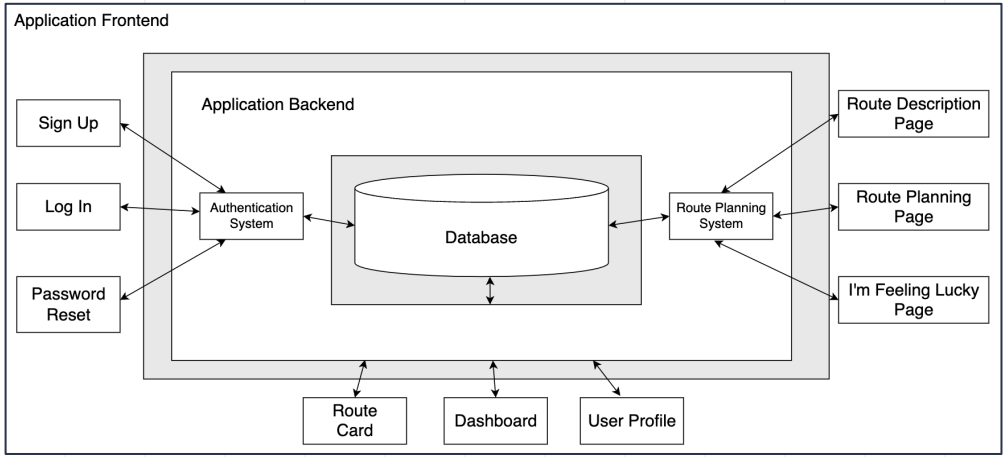


BACKEND

DATABASE

Datasets used

System Architecture Diagram

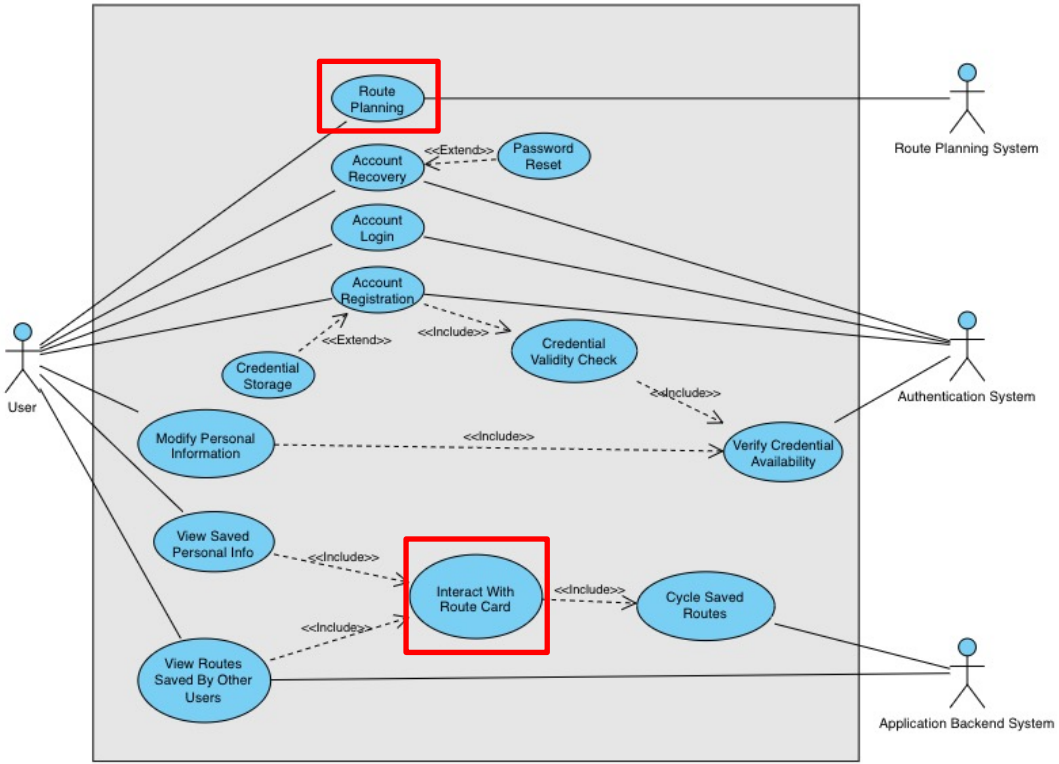


Features

High level structure of web application

Loose coupling, high cohesion: Classes are split into different "layers" and "layers" can only communicate with neighbouring "layers"

Use Case Diagram



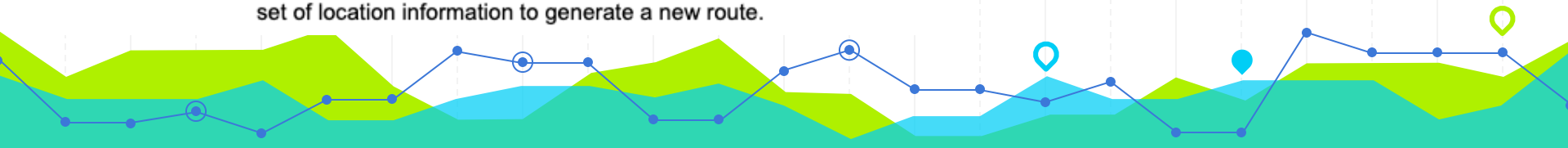
Use Case: Route Planning

Functional Requirements Traceability

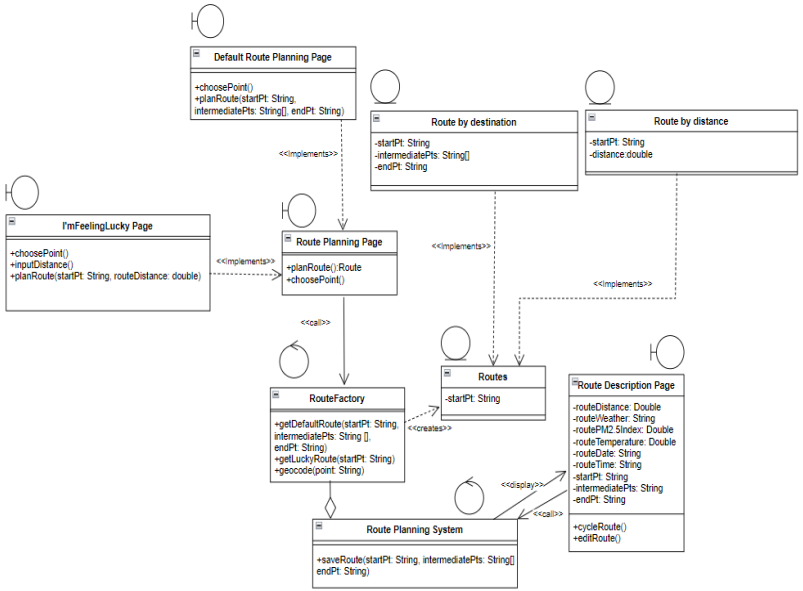
c. The website must consist of a Route Plotting Feature
   i. The website provides capabilities to plot cycling routes only.
   ii. To use the Route Plotting Feature, users must input location information.
      1. The user must input the start destination
      2. The user must input the end destination
      3. The user is given the option to add extra destinations along the route.
   iii. After the user provides all the location information, the system will return the proposed route.
      1. The proposed route must be represented on a map
      2. There must be a clear indication of the whole cycling route, including the start and the end point.
      3. The route distance must be indicated.
      4. The PM2.5 index for the route must be indicated.
      5. The weather data for the route must be indicated.
   iv. Based on the proposed route returned, the user must be able to choose to "select" or "reject" the route.
      1. If the user selects the route, an image of the route is generated.
      2. If the user rejects the route, the user must be able to input a new set of location information to generate a new route.

Main Points:

- User can add extra destinations
- The weather and route information will be shown
- User can choose to select or reject the route



Use Case: Route Planning



Boundary Classes

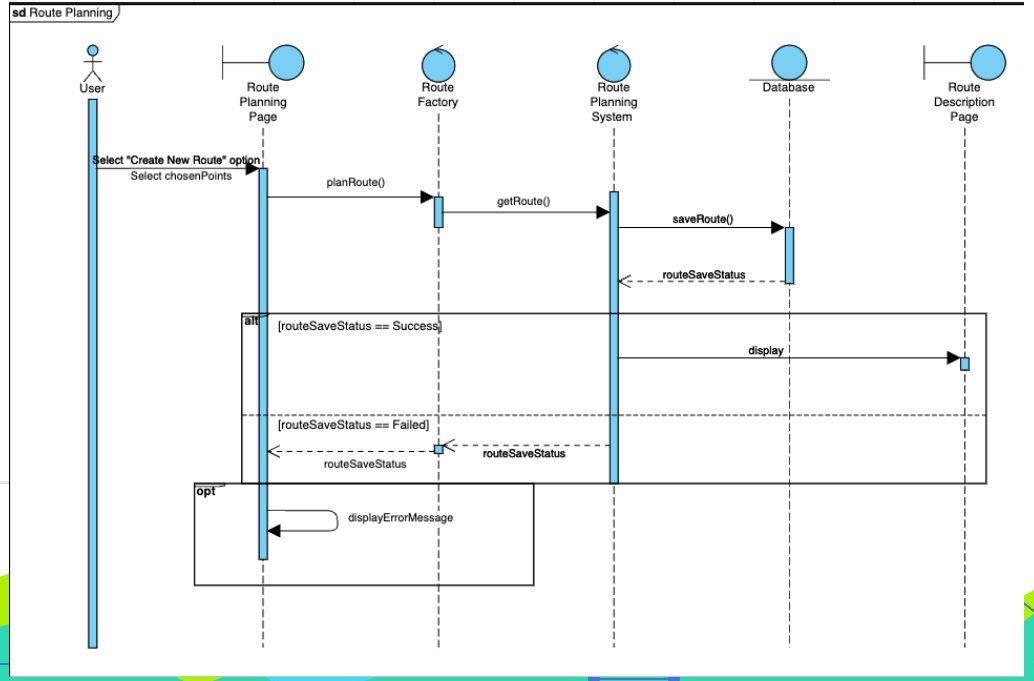
- Route planning pages
- Route description page

Control Classes

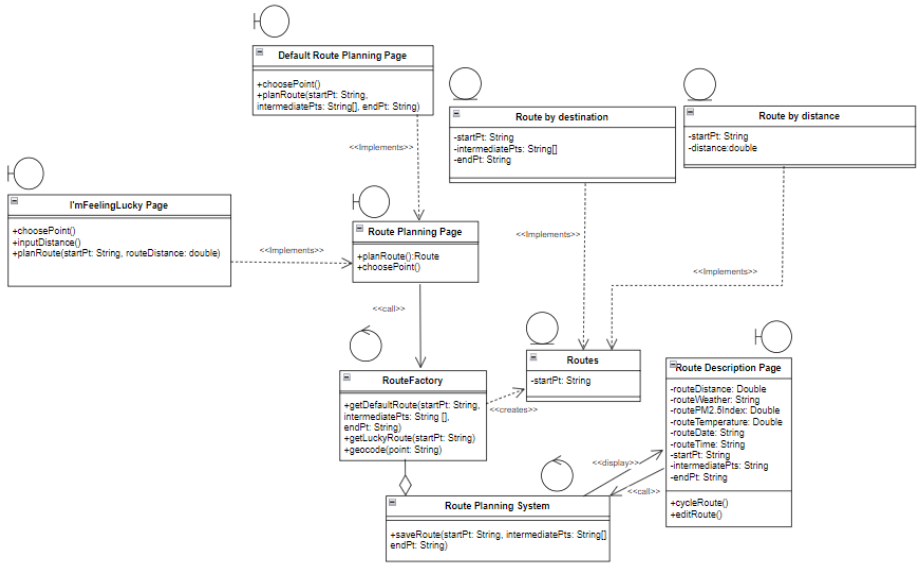
- Route planning system
- RouteFactory

Entity Classes

- Route by Destination
- Route by Distance
- Routes



Design Patterns for Route Planning Function



Factory Design + Strategy Design

- Generate Class only on Runtime (Dynamic loading)
- Increases robustness and performance
- Ease in extending or adding new subclasses

Blackbox Testing for Route Planning Function

Class Method: getDefaultRoute()

Inputs:

1. All Valid Inputs
2. Empty set of intermediate points
3. Invalid endPt (outside Singapore)
4. Invalid startPt (outside Singapore)

Equivalence classes with invalid endPt and startPt inputs were tested.

	startPt	intermediatePts	endPt	Expected Output	Actual Output
1	"1.3406787810831653,103.80102557666015"	["1.3355303441379147,103.8102952910156 2", "1.3329561216181542,103.81716174609375"]	"1.3370748763558524,103.82643146044921"	API returns 200 OK Response	API returns 200 OK Response
2	"1.3406787810831653,103.80102557666015"	[]	"1.3370748763558524,103.82643146044921"	API returns 200 OK Response	API returns 200 OK Response
3	"1.3406787810831653,103.80102557666015"	[]	"1.4697847651217744,103.55368419713508"	API returns 400 Bad Request	API returns 400 Bad Request
4	"1.4697847651217744,103.55368419713508"	[]	"1.3406787810831653,103.80102557666015"	API returns 400 Bad Request	API returns 400 Bad Request

Class Method: getLuckyRoute()

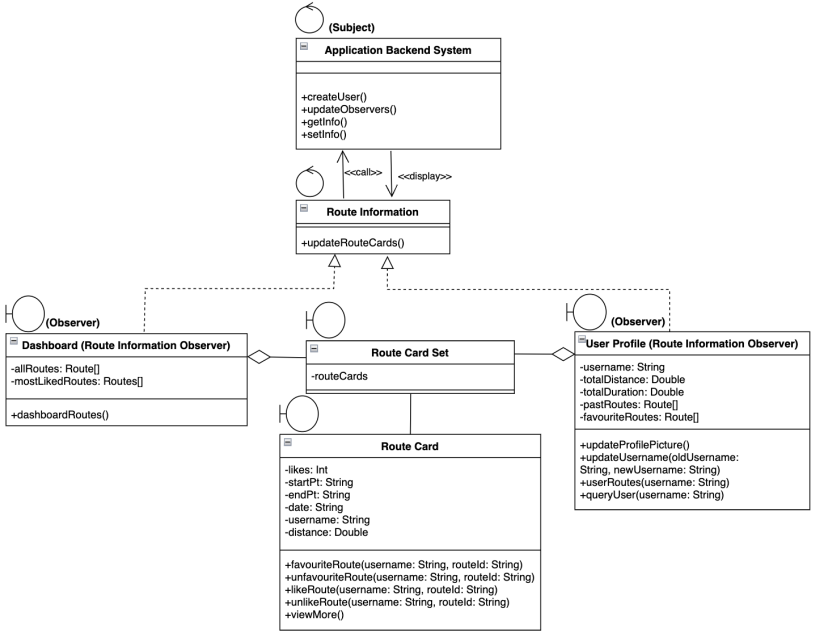
Inputs:

1. All Valid Inputs, with just-above routeDistance input of 1, and just-below startPt of "1.408,103.925"
2. Invalid on-the-boundary routeDistance input of 0
3. Invalid just-below routeDistance input of -1
4. Valid on-the-boundary startPt of "1.4089433,103.9255495" (Singapore boundary)
5. Invalid just-above startPt of "1.41,103.93" (outside Singapore boundaries)

Equivalence classes with invalid startPt and routeDistance inputs were tested, and boundary value testing was applied for routeDistance and startPt inputs

startPt	routeDistance	Expected Output	Actual Output
"1.408,103.925"	1	API returns 200 OK Response	API returns 200 OK Response
"1.408,103.925"	0	API returns 400 Bad Request	API returns 400 Bad Request
"1.408,103.925"	-1	API returns 400 Bad Request	API returns 400 Bad Request
"1.4089433,103.9255495"	1	API returns 200 OK Response	API returns 200 OK Response
"1.41,103.93"	1	API returns 400 Bad Request	API returns 400 Bad Request

Use Case: Interact with Route Card

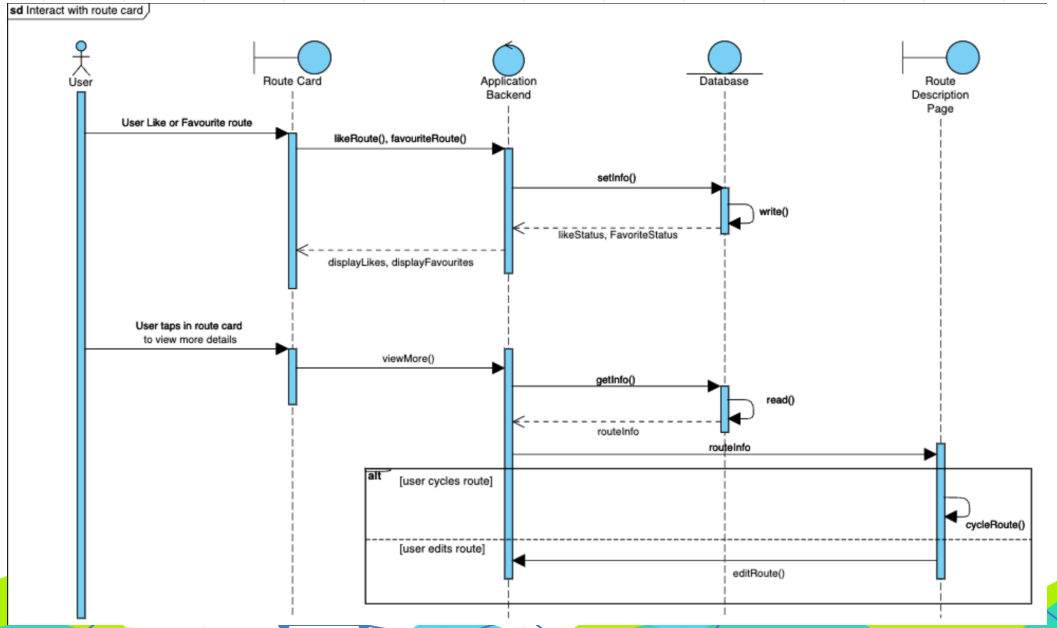


Boundary Classes

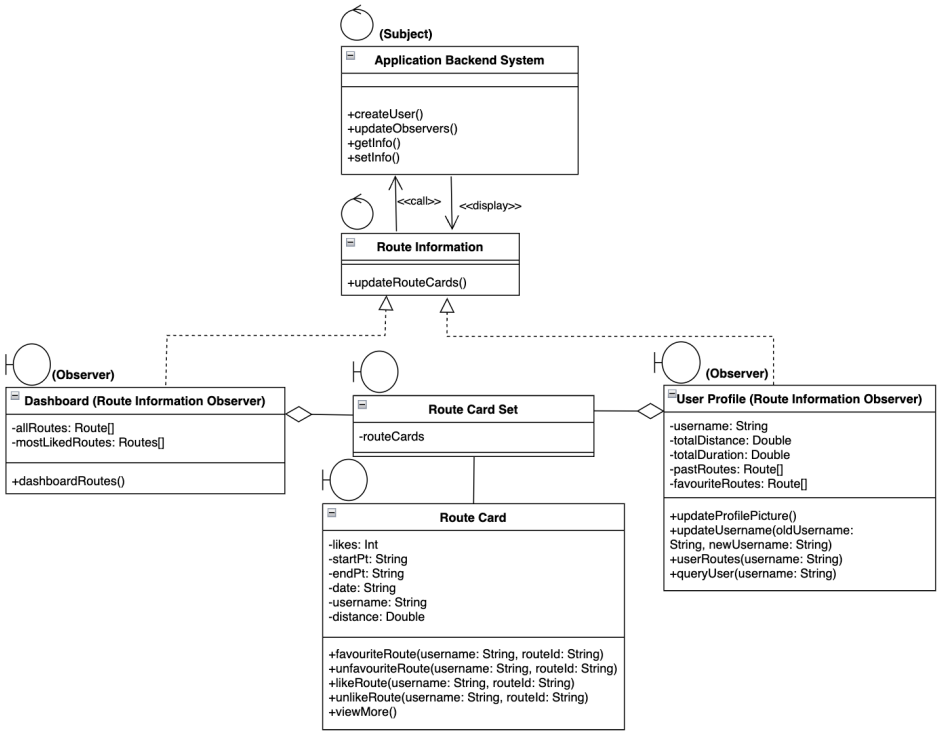
- Route Card Set
- Dashboard
- User Profile

Control Classes

- Application Backend
- Route Information



Design Patterns for Interact with Route Card

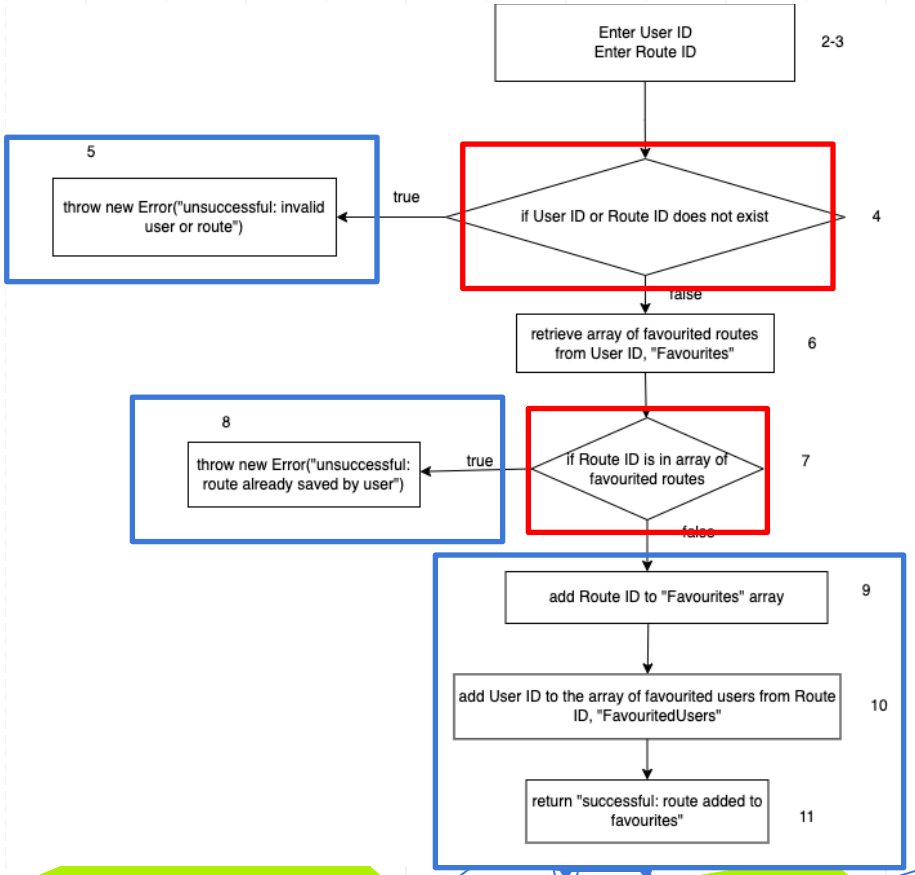


Observer Pattern

Updates many observers
When the subject changes

Loose coupling between observers and subjects

Whitebox Testing for favouriteRoute()



- 2 Decision points (Red)
- 3 Basis Paths (Blue)

3 Basis Paths:

1. Valid User ID, Route ID & Route ID not in “Favourites” array
2. Either User ID or Route ID is invalid
3. Valid User ID, Route ID & Route ID is already in “Favourites” array

Testing for favouriteRoute()

	User ID	Route ID	Expected Output	Actual Output
1	test	3TbKc61tF5rFMoY1Wq1g	Successful: route added to favourites	Successful: route added to favourites
2a	test	123	Unsuccessful: invalid user or routes	Unsuccessful: invalid user or routes
2b	user	3TbKc61tF5rFMoY1Wq1g	Unsuccessful: invalid user or routes	Unsuccessful: invalid user or routes
3	test	8EOn7GNDSXORNQXBvDfQ	Unsuccessful: route already saved by user	Unsuccessful: route already saved by user

Good software engineering practice: weekly scrum meetings

Thank You!