

Power Up!

Power Up!

By Power Rangers

Team Members:

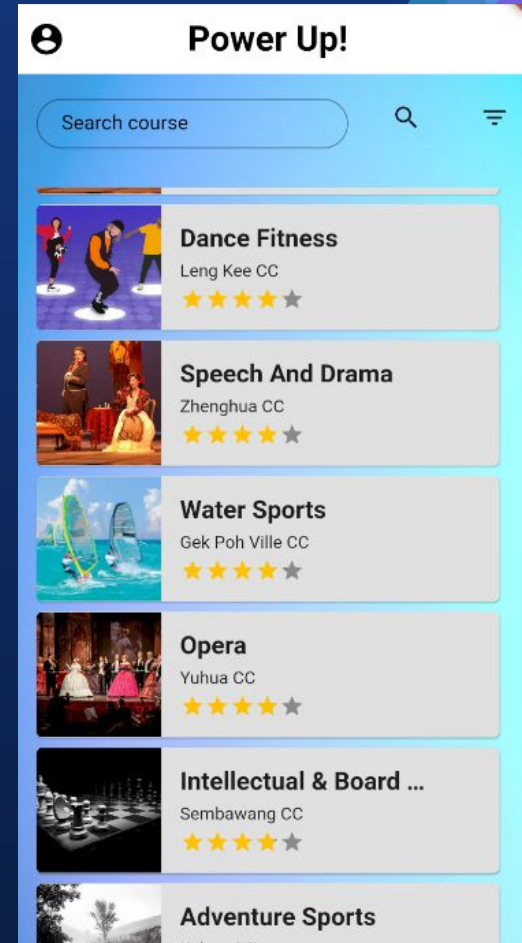
Jaw Li Sheng, Jessica Halim, Ong Yi
Xiong, Chua Wen Qing, Syed Anas Majid,
Tang Kai Wen, Alvin

What Is Power Up?

Power Up is a mobile application that aims to connect people who are seeking to develop a skill, interest or hobby to vendors that provide such lessons

Our Target Audience

We aim to target age group above the age of 7 from the young to the elderly



Use Case Diagram

User

User Registration → Email Validation

Search

Filter

Order By

Withdraw from a Course

Register for a Course

Bookmark Favourite Courses

Retrieve lists of favourite/registered courses

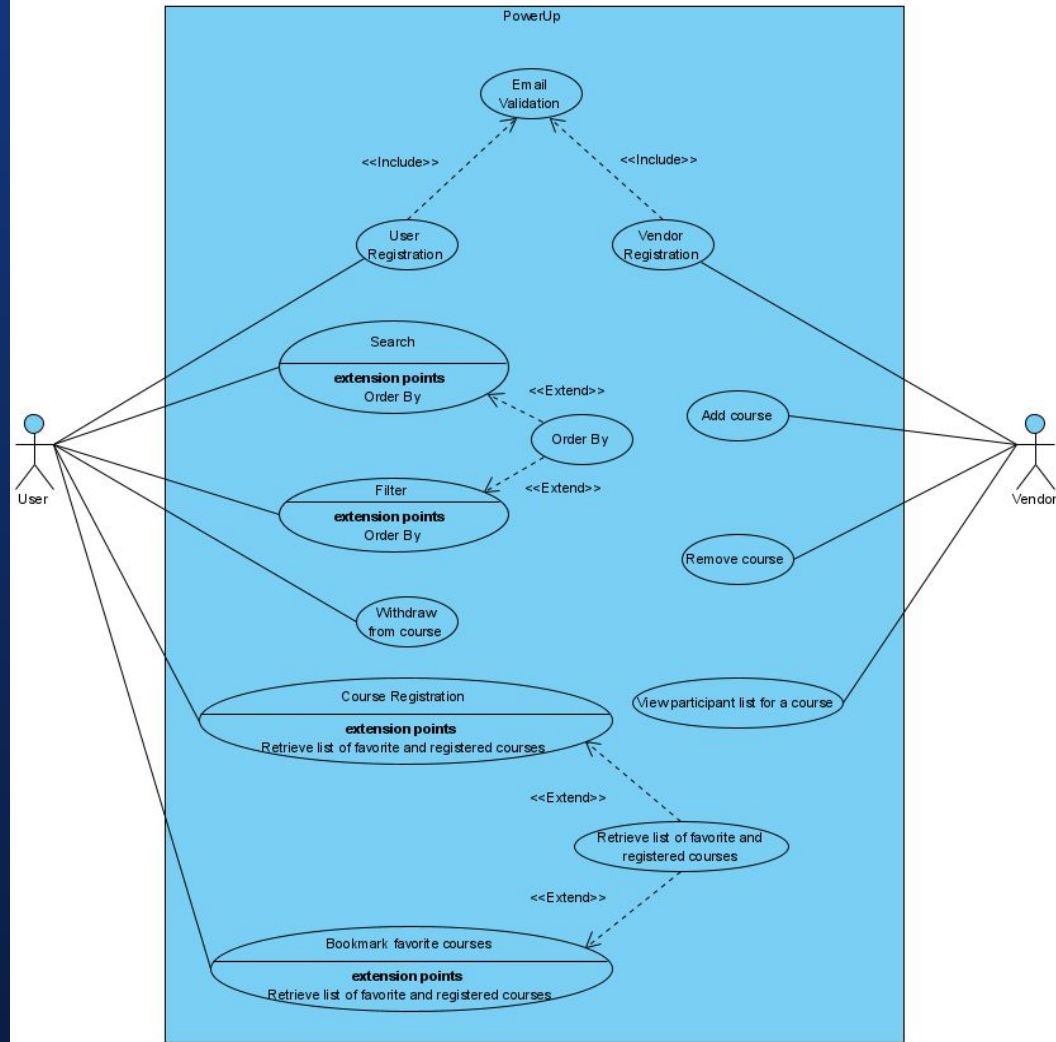
Vendor

Vendor Registration → Email Validation

Add Course

Remove Course

View Participant List (for each course)



DEMO



TABLE OF CONTENTS

1. **System Design Overview**
An overview of the application's architecture

2. **Good Practices & Design**
Software engineering principles put into practice

3. **Traceability**
Consistency of requirements throughout development

4. **Future Improvements**
Possible areas for improvement.



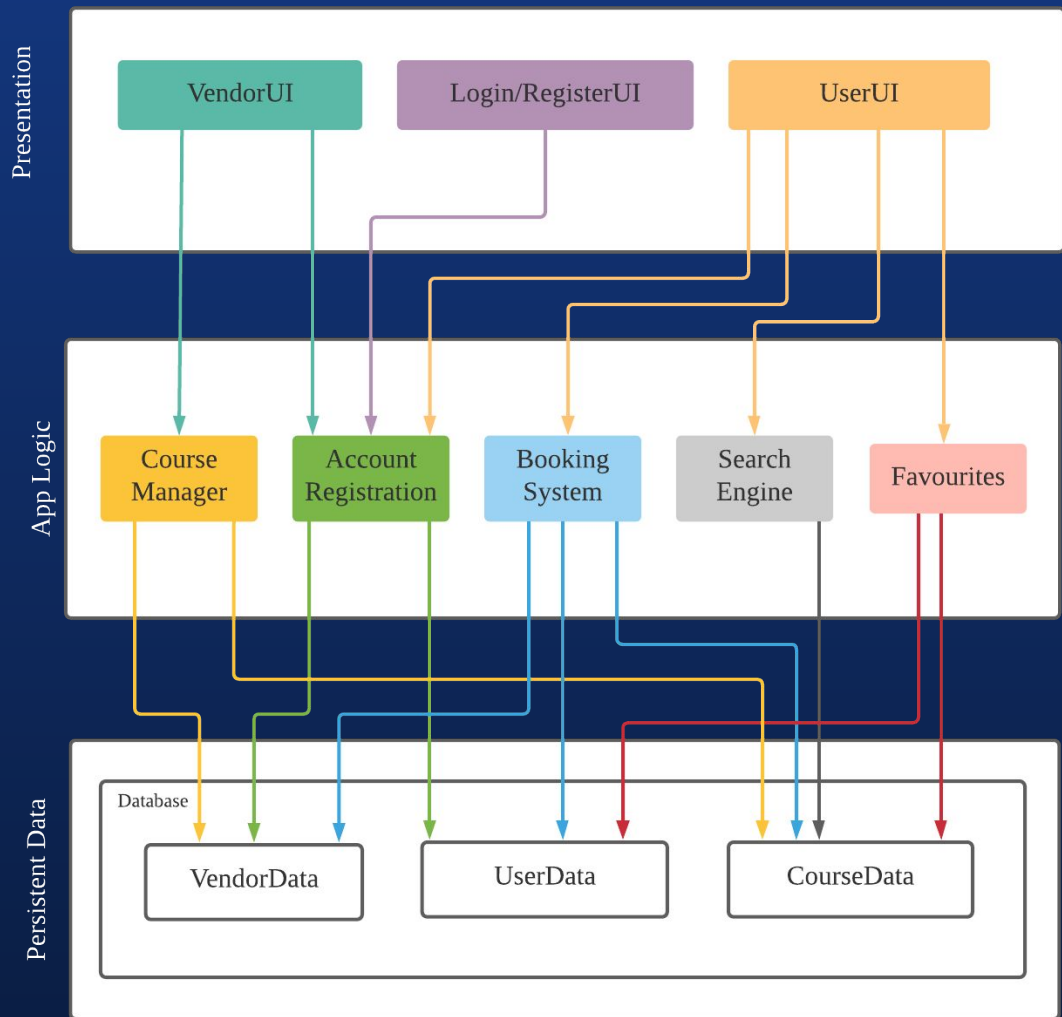
01

SYSTEM DESIGN OVERVIEW



1. SYSTEM DESIGN OVERVIEW

- Layered architecture
- Clear communication structure/hierarchy





02

GOOD PRACTICES & DESIGN



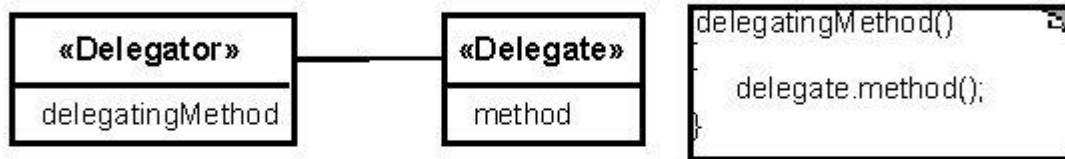
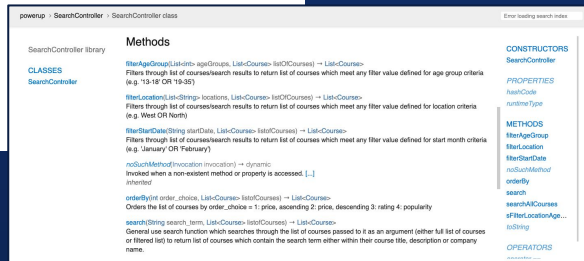
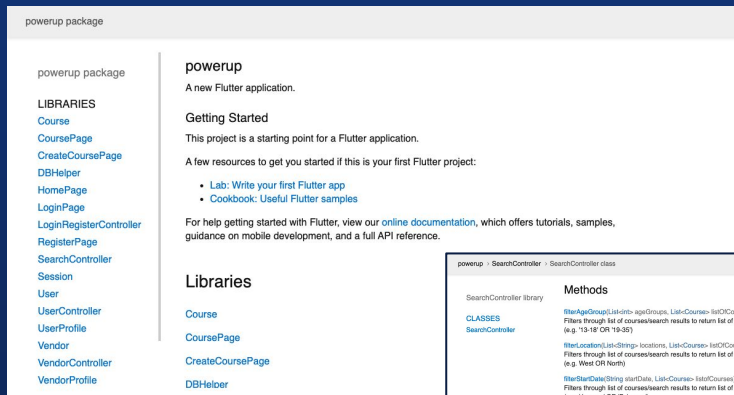
2. GOOD PRACTICES & DESIGN

1. *Maintainability*

- Separation of concerns; modularity
- Documentation

2. *Extensibility*

- Delegation (instead of implementation inheritance)



2. GOOD PRACTICES & DESIGN

Design Pattern: **Singleton Pattern**

Singleton: ensures that a class has only one instance, and provides a global access point to this instance.

```
/// Singleton
static DBHelper single_instance = null;
static DBHelper getInstance()
{
    if (single_instance == null)
        single_instance = new DBHelper();
    return single_instance;
}
```

The only way to obtain an instance of the singleton class.

Memory will be allocated only on the first call to the "single_instance" property.

get the reference of a preconfigured instance from a working class.

```
class LoginRegisterController{
    DBHelper dbHelper = DBHelper.getInstance();
}
```

Singleton Benefits:

- Resource-friendly
- Improves performance
- Better Scalability



2. GOOD PRACTICES & DESIGN

Design Pattern: **Factory Pattern**

- Flutter support for singleton - **factory** constructor



Factory Benefits:

- Encapsulation of object creation
- Extensibility
- Easy to change object creation logic





2. GOOD PRACTICES & DESIGN

Single Responsibility Principle (SRP)

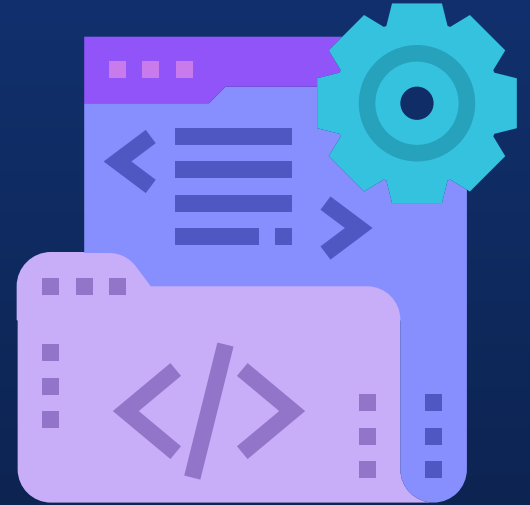
- Classes are created to focus on solely one responsibility (LoginRegisterController handles *only* checks with regards to user inputs for login or registration)
- Benefits: Testing is easier with each class having fewer test cases, lower coupling as lesser functionality in a single class will have fewer dependencies.





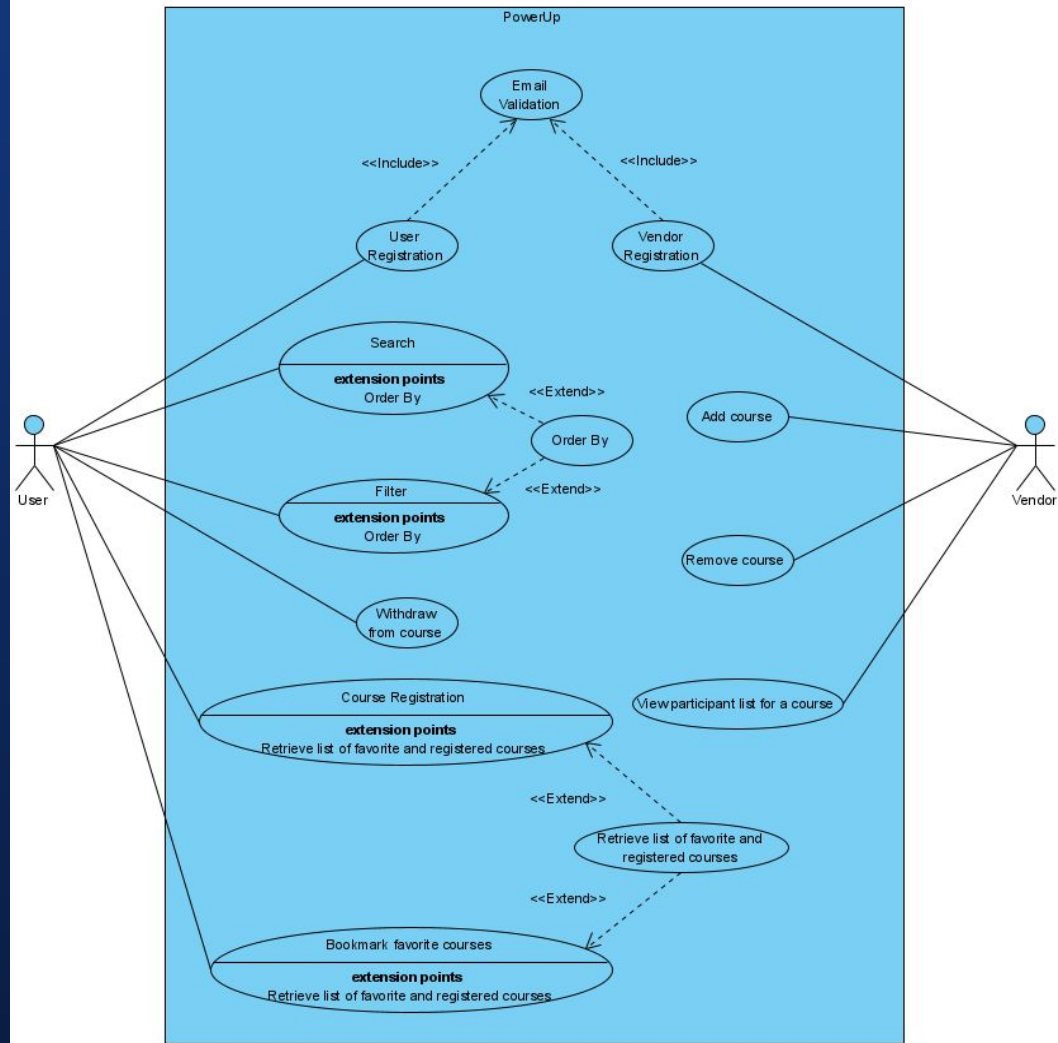
03

TRACEABILITY



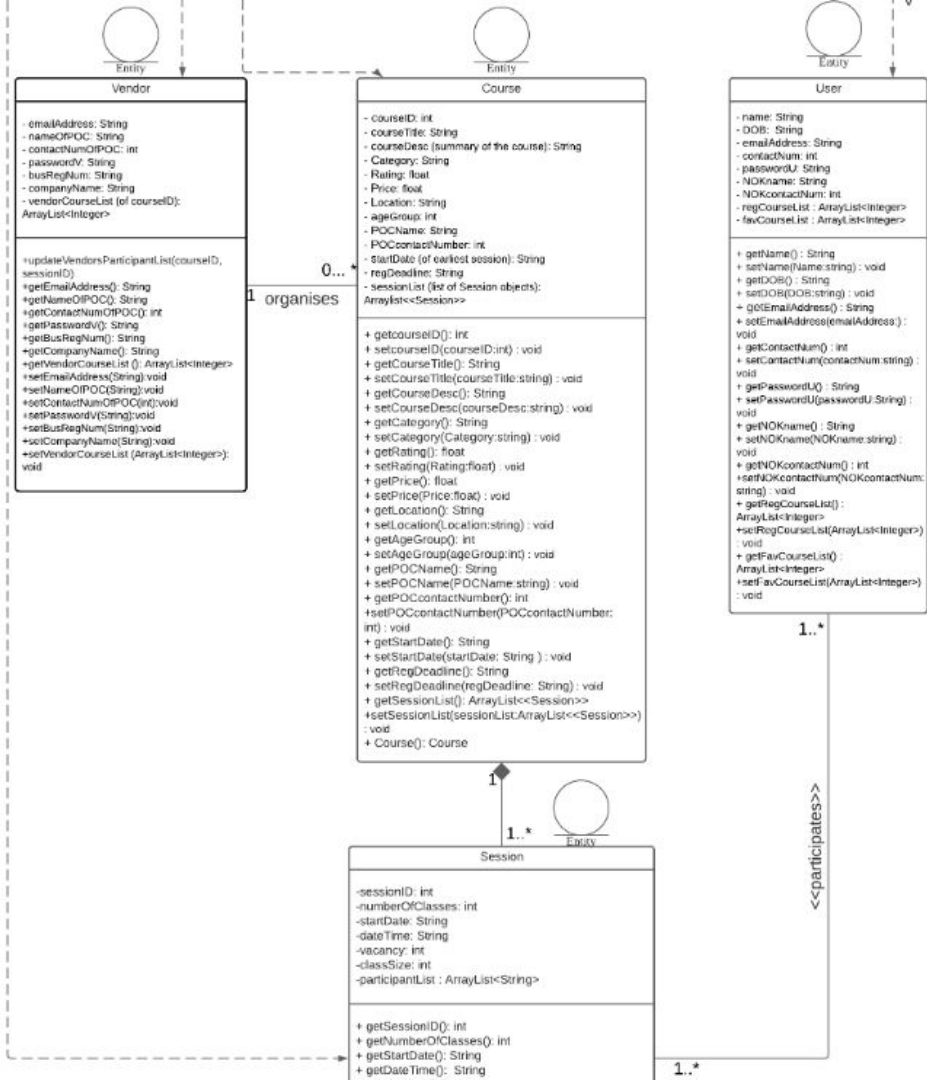
3. TRACEABILITY

- Use Case: User Register For Course



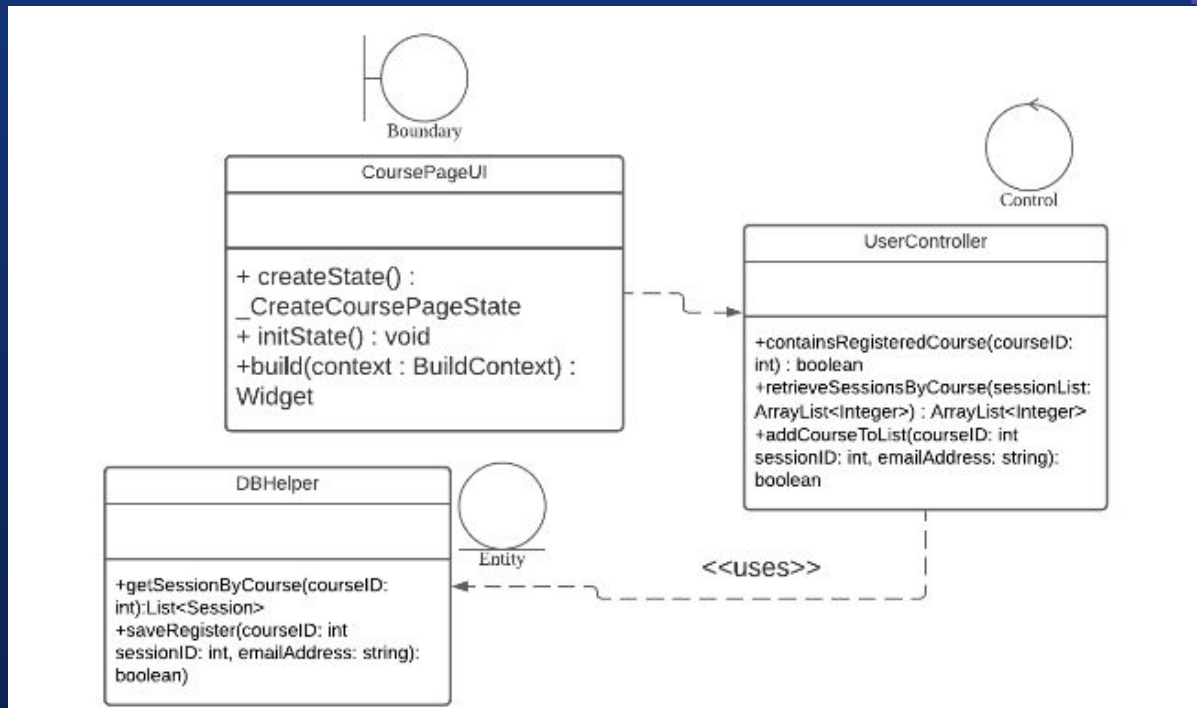
Use Case: User Register For Course

Entity Classes



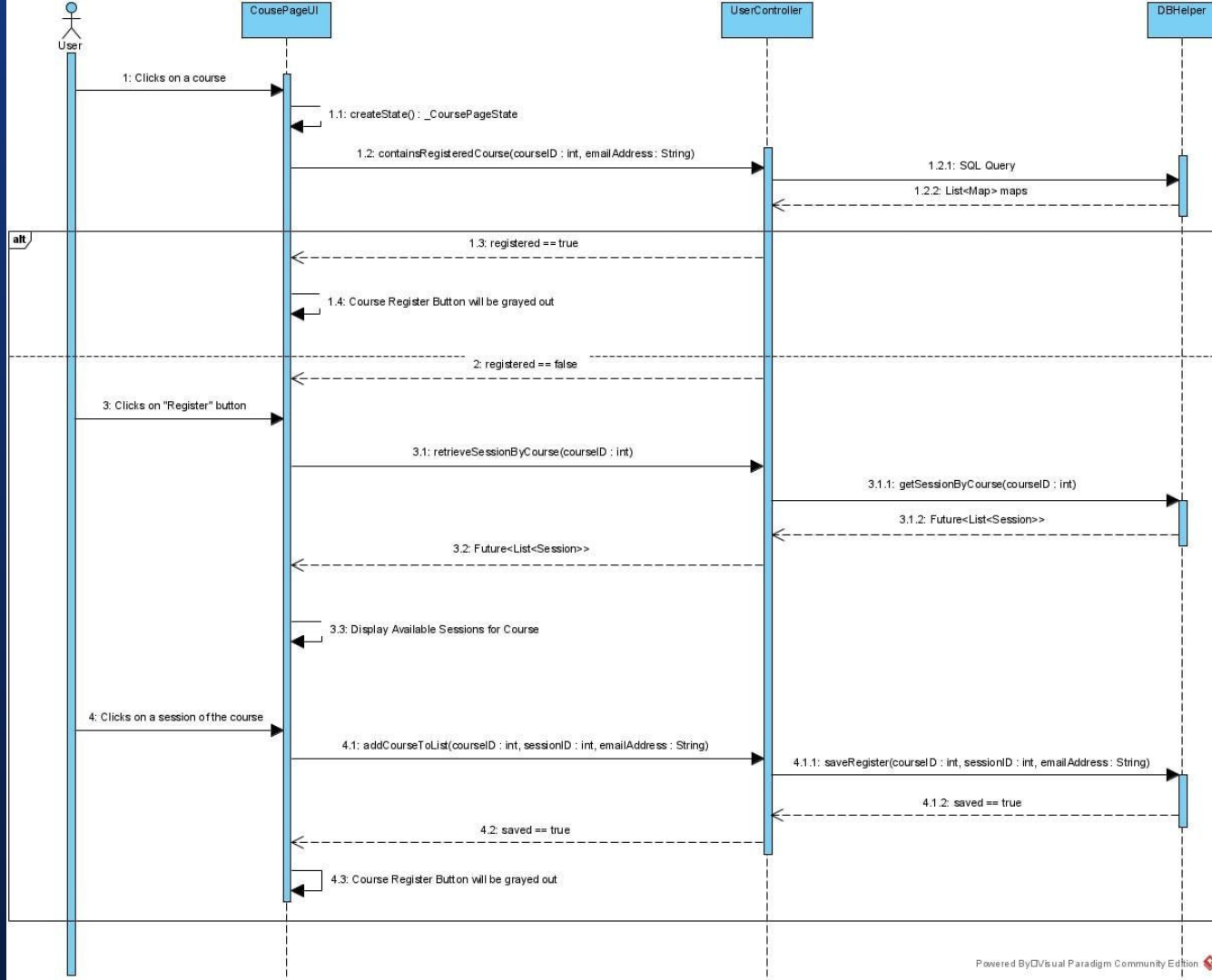
Use Case: User Register For Course

Boundary, Control, Entity Classes



Use Case: User Register For Course

Sequence Diagram





Black Box Testing(User Registration)

	Last Name	First Name	Date Of Birth	Contact Number	Email Address	Password	Confirm Password	Name Of NOK	Contact number of NOK	Expected Output	Actual Output
Wrong Date format	Boh	Ben	12/12/19989	98769876	ben@gmail.com	Hello123#	Hello123#	Bob boh	91231234	"Wrong Data Format"	"Wrong Data Format"
Invalid Number	Boh	Ben	12/12/2000	12345678	ben@gmail.com	Hello123#	Hello123#	Bob boh	91231234	"Contact number is invalid.Please try again"	"Contact number is invalid.Please try again"
Invalid Password	Boh	Ben	12/12/2000	98769876	ben@gmail.com	123	123	Bob boh	91231234	"Password should contain at least 8 characters with at least 1 Uppercase,1 lowercase,1 digit"	"Password should contain at least 8 characters with at least 1 Uppercase,1 lowercase,1 digit"



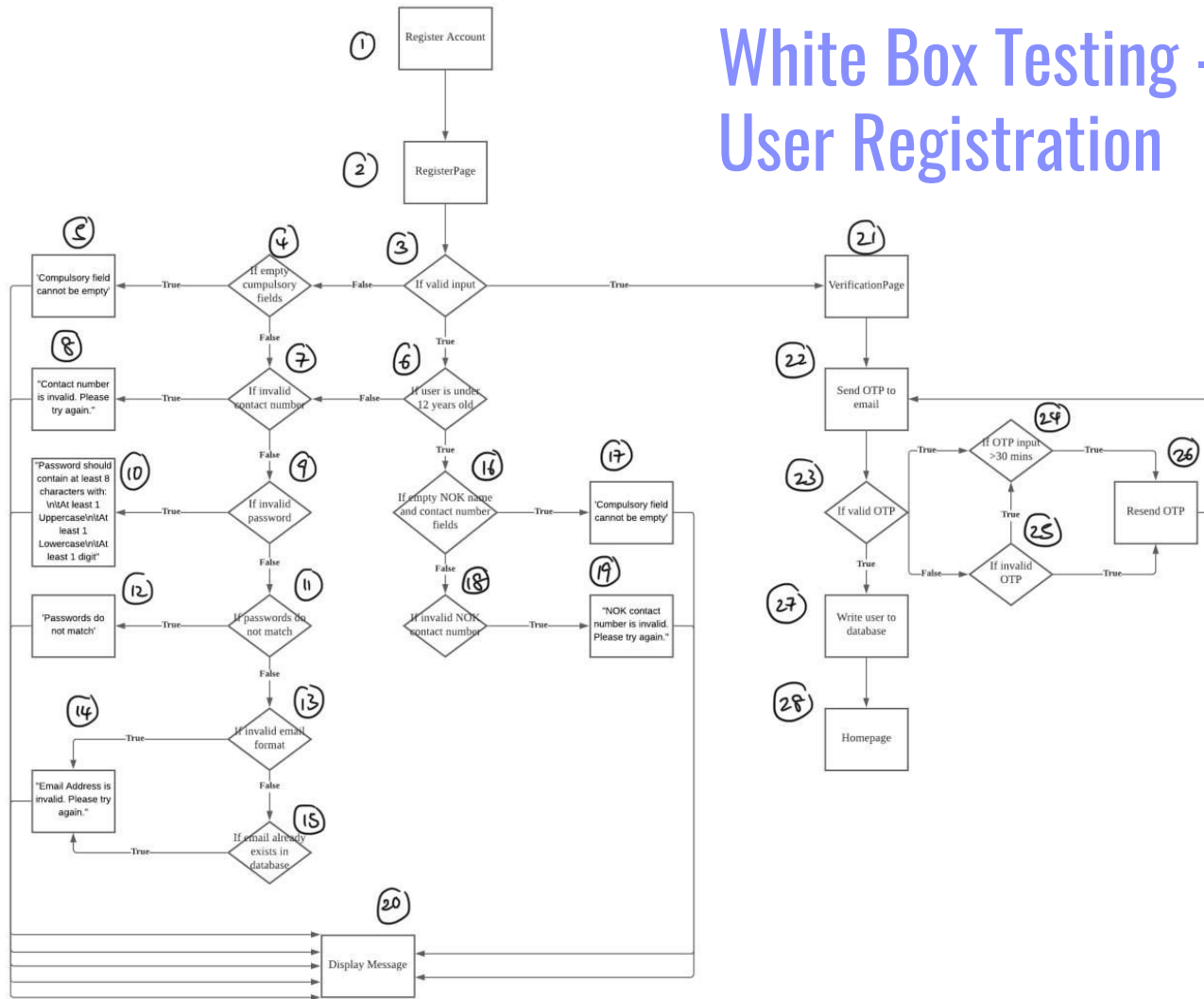


Black Box Testing(User Registration)

	Last Name	First Name	Date Of Birth	Contact Number	Email Address	Password	Confirm Password	Name Of NOK	Contact number of NOK	Expected Output	Actual Output
Password Do not Match	Boh	Ben	12/12/2000	98769876	ben@gmail.com	Hello123#	Hello123	Bob boh	91231234	"Passwords do not match"	"Passwords do not match"
Unsuccessful Verification	Boh	Ben	12/12/2000	98769876	ben@gmail.com	Hello123#	Hello123#	Bob boh	91231234	User unable to register	User unable to register
Successful verification	Boh	Ben	12/12/2000	98769876	ben@gmail.com	Hello123#	Hello123#	Bob boh	91231234	Homepage	Homepage
Incomplete Field	Boh	Ben		98769876	ben@gmail.com	Hello123#	Hello123#	Bob boh	91231234	"Compulsory field cannot be empty"	"Compulsory field cannot be empty"

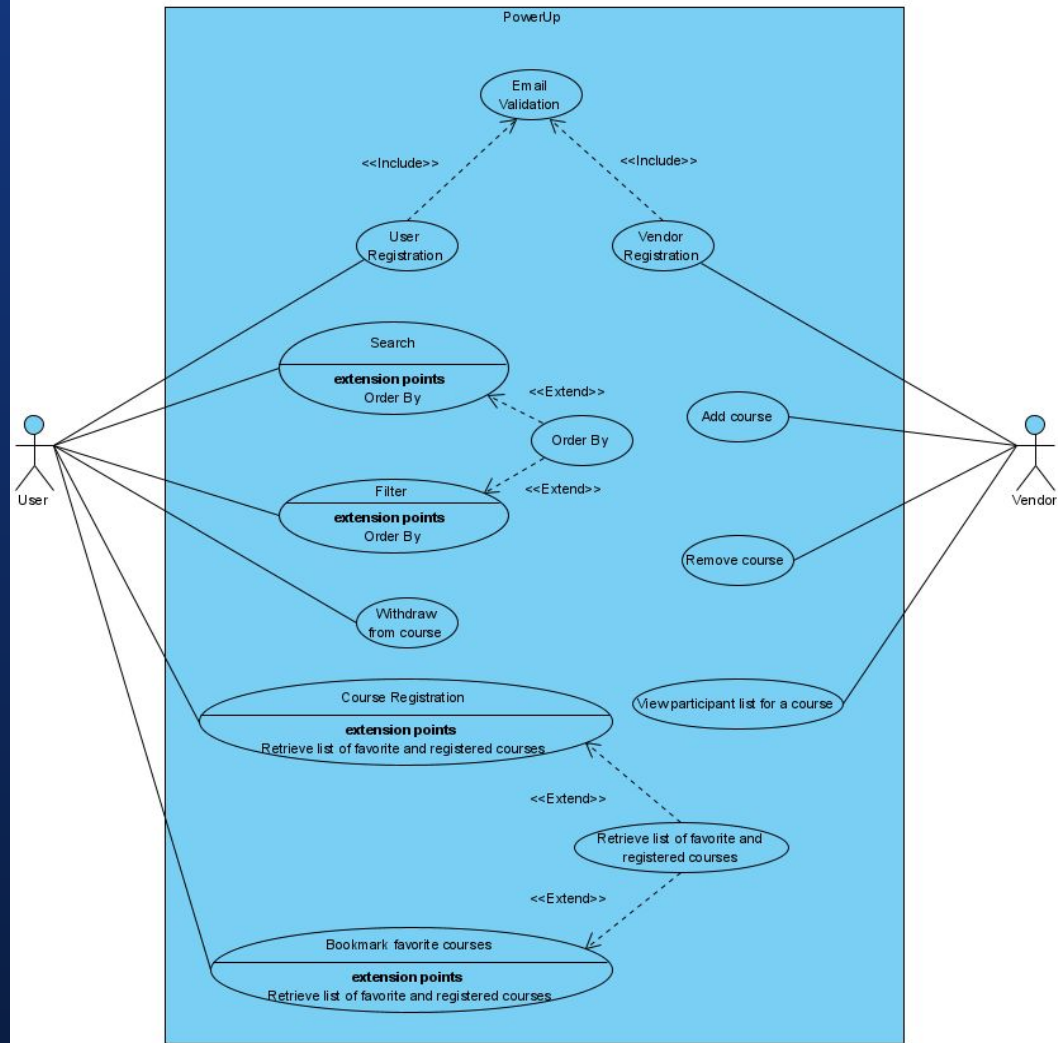


White Box Testing - User Registration



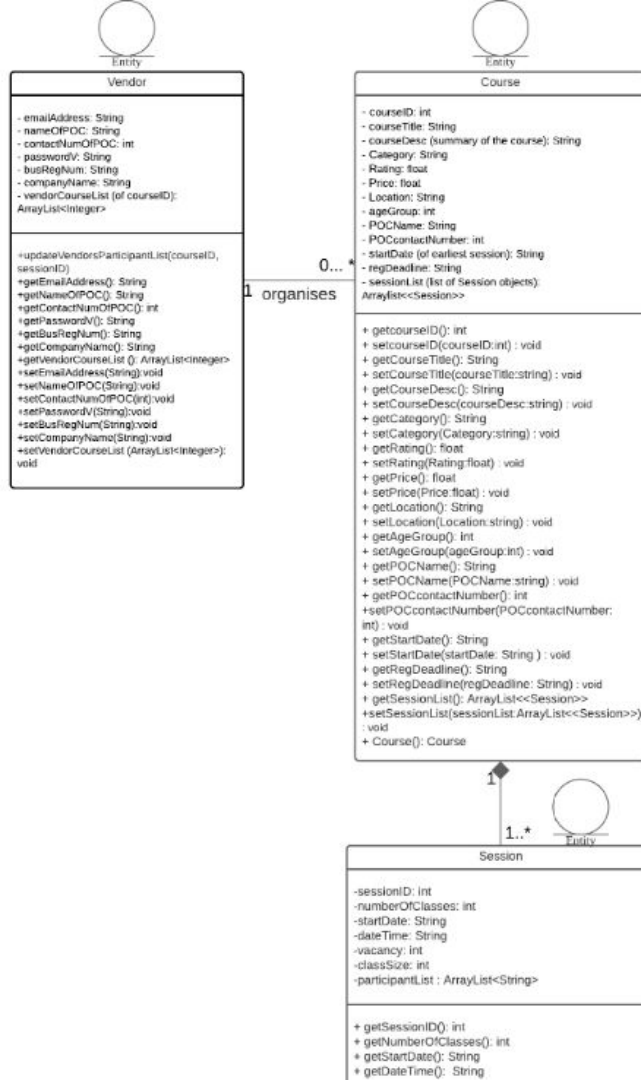
3. TRACEABILITY

Use Case: Add Course (Vendor)



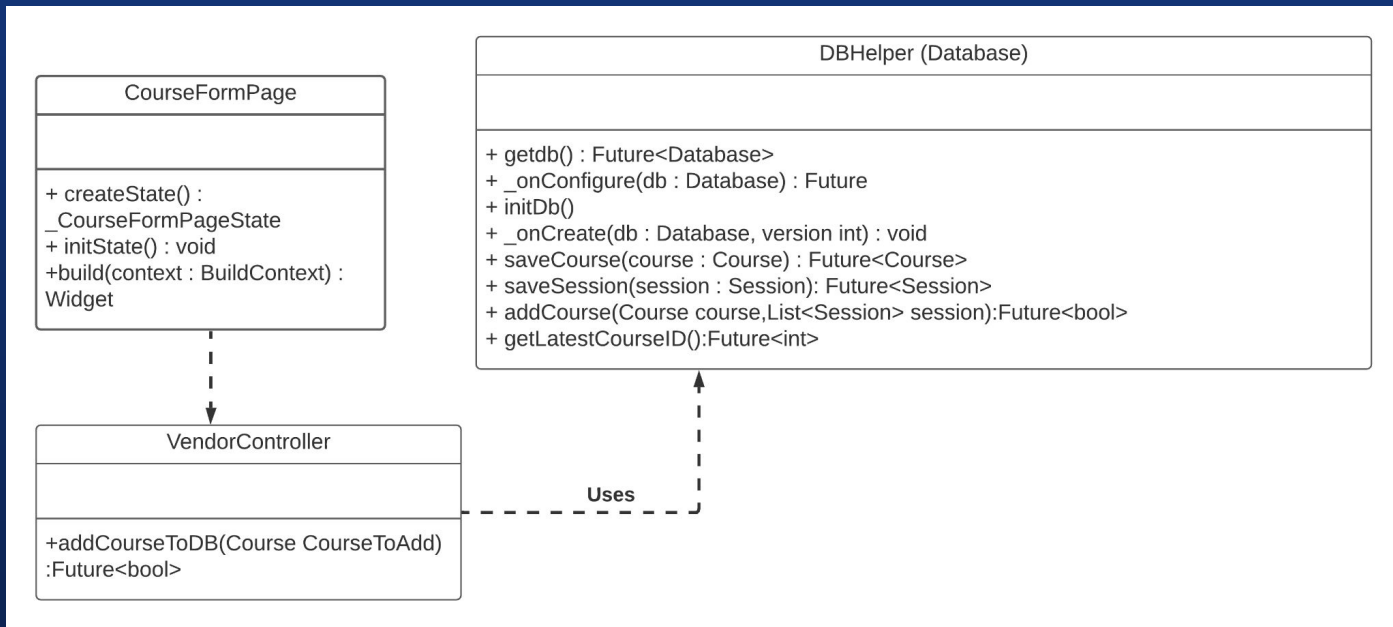
Use Case: Add Course (Vendor)

Entity Classes



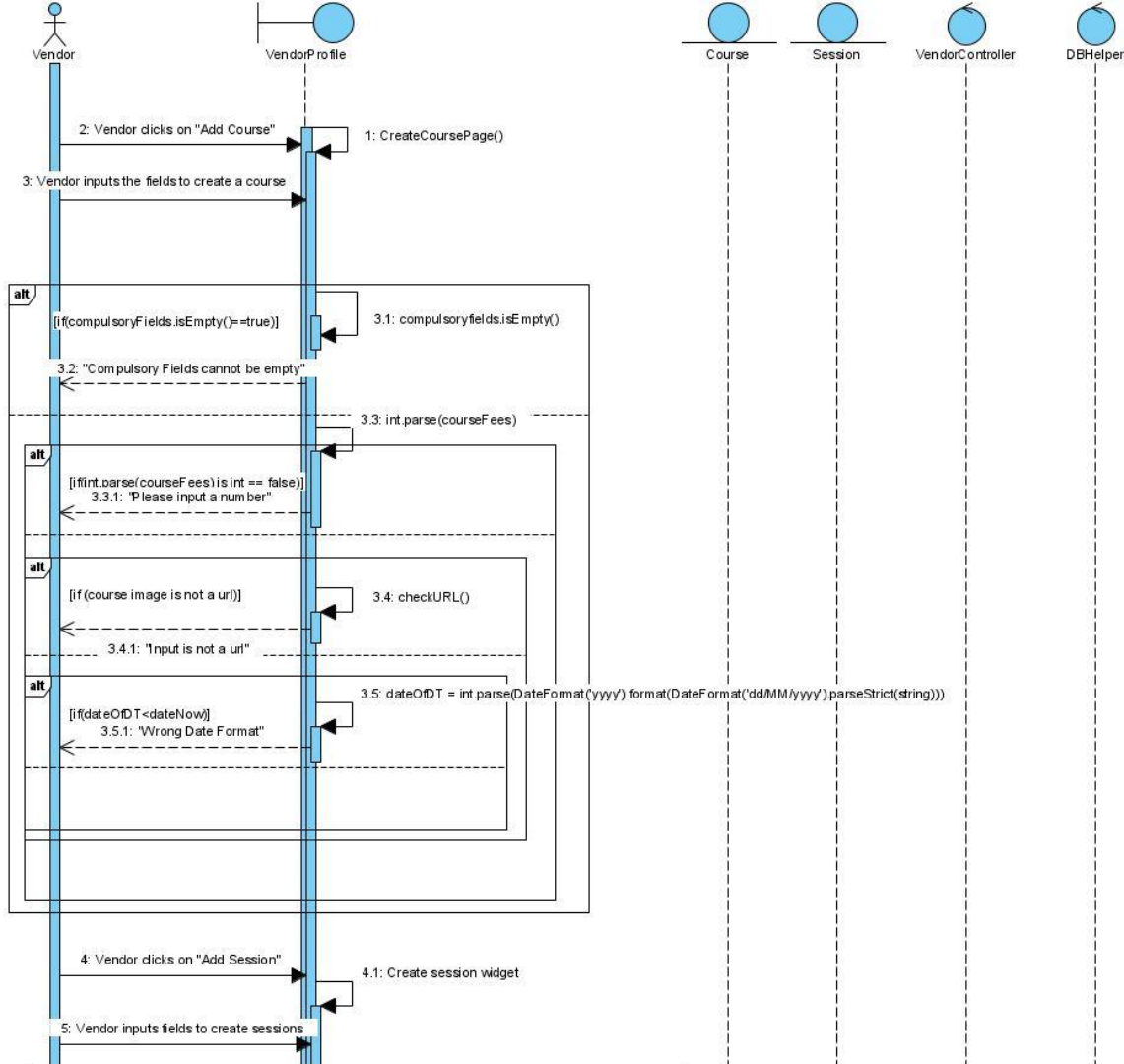
Use Case: Add Course (Vendor)

Boundary, Control, Classes



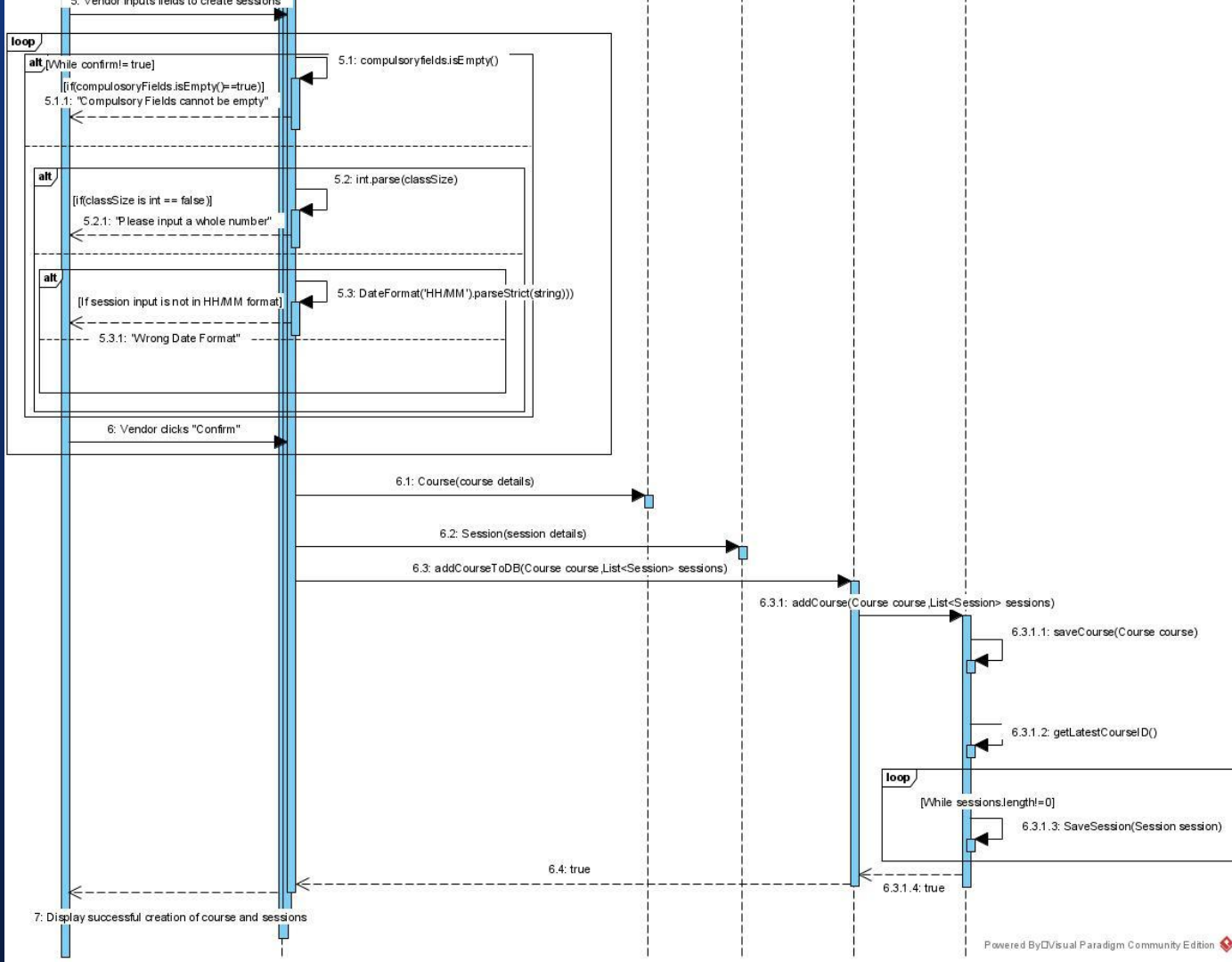
Use Case: Add Course (Vendor)

Sequence Diagram (Part 1)



Use Case: Add Course (Vendor)

Sequence Diagram (Part 2)



3. TRACEABILITY - Use Case: Add Course (Vendor)



Black box testing (Main Course Details)

Title	Description	Location	Url Image	Age Group	Fees	Reg Deadline	Start Date	Expected Output	Actual = Expd?
If any of the above fields are empty (or incomplete)								'Compulsory field cannot be empty'	✓
Crafts 101	Welcome to the world of crafts!	26 Lorong Avenue, Singapore 989777	'N.A.'	13-18	20.30	20/10/2021	20/11/2021	'Input is not a URL.'	✓
Crafts 101	Welcome to the world of crafts!	26 Lorong Avenue, Singapore 989777	https://icatcare.org/app/uploads/2018/07/Thinking-of-getting-a-cat.png	13-18	'Not sure'	20/10/2021	20/11/2021	'Please input a number'	✓
Crafts 101	Welcome to the world of crafts!	26 Lorong Avenue, Singapore 989777	https://icatcare.org/app/uploads/2018/07/Thinking-of-getting-a-cat.png	13-18	20.30	'20.10.2021' or 'Not sure' or '20/10/21'	20/11/2021	'Wrong date input'	✓
Crafts 101	Welcome to the world of crafts!	26 Lorong Avenue, Singapore 989777	https://icatcare.org/app/uploads/2018/07/Thinking-of-getting-a-cat.png	13-18	20.30	20/10/2021	'20.11.2021' or 'Not sure' or '20/11/21'	'Wrong date input'	✓





3. TRACEABILITY - Use Case: Add Course (Vendor)

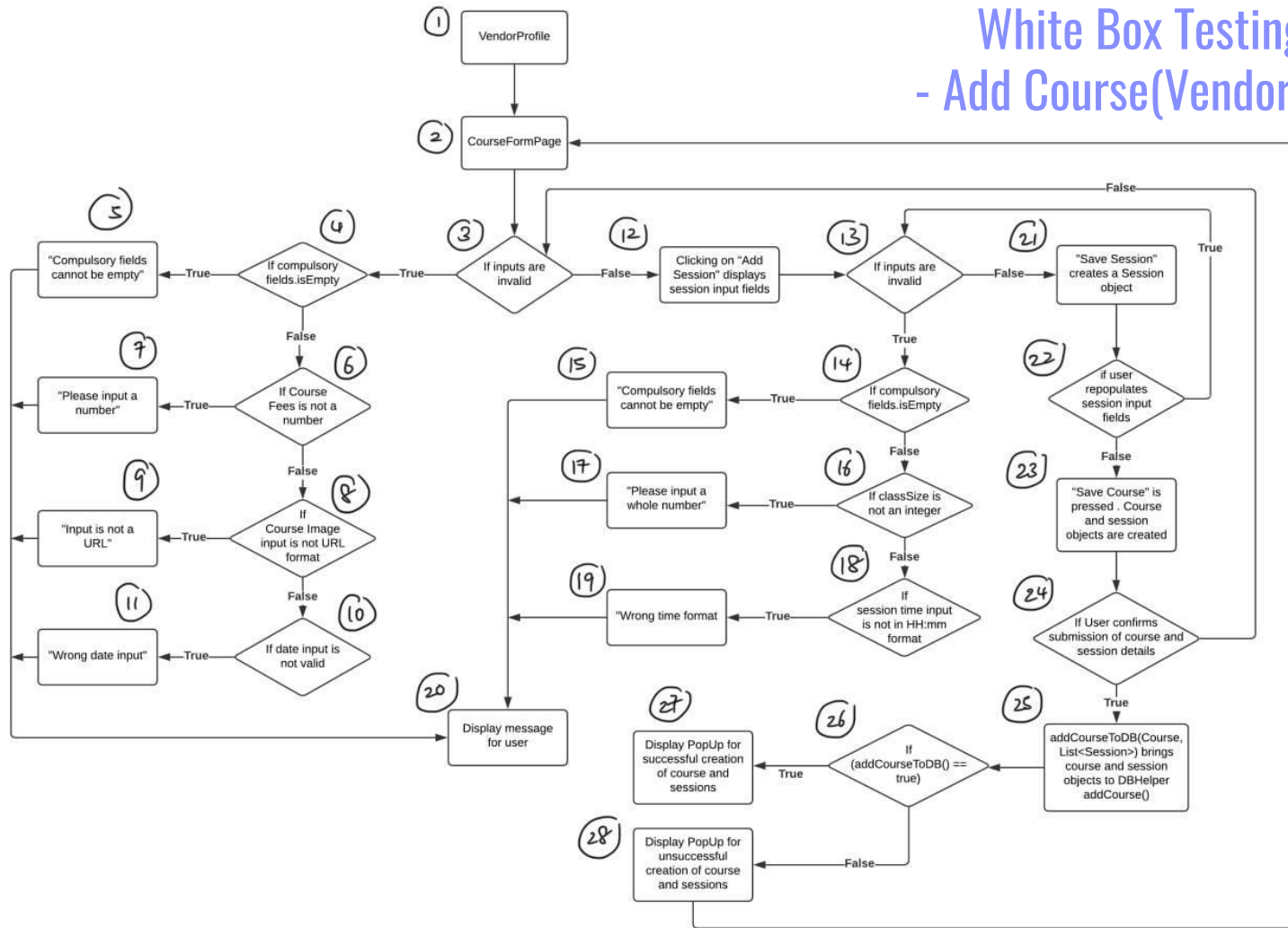
Black box testing (Session Widget)

Session Count	Class Size	Start Date	Start Time	Expected Output	Actual O/P = Expected?
3	10.3	20/11/2021	10:30	'Please only input whole numbers.'	✓
3	10	'20.11.2021' or 'Not sure' or '20/11/21'	10:30	'Wrong date format'	✓
3	10	20/11/2021	'24:40' or '8:61' or 'Not sure' or ':40' or '9'	'Wrong date format'	✓



White Box Testing

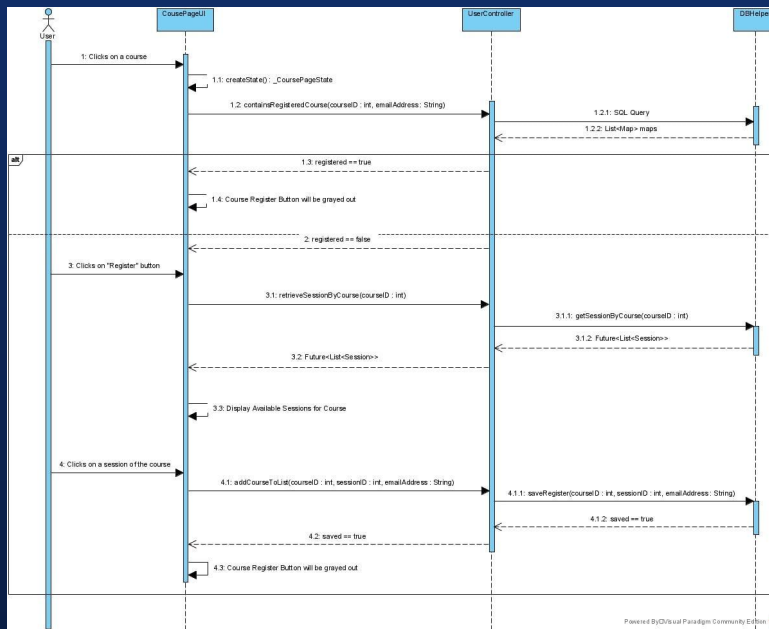
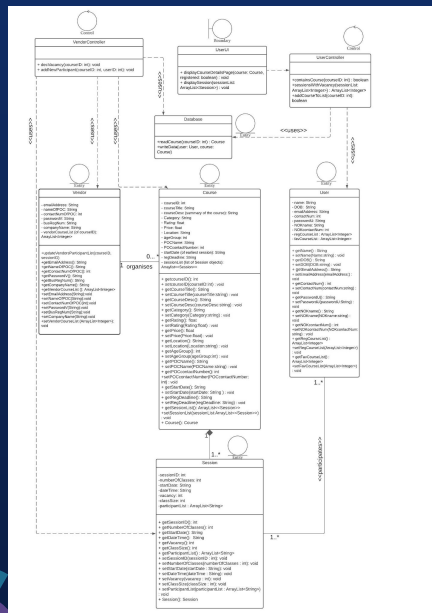
- Add Course(Vendor)





3. TRACEABILITY - Use Case: User Register For Course

Do 1-2 use case diagram here and show relevant class diagrams and sequence diagrams for the use case, mention the good design we have applied, how we have implemented them, what testing we performed





Further white box and black box testing





Black Box Testing(Login)

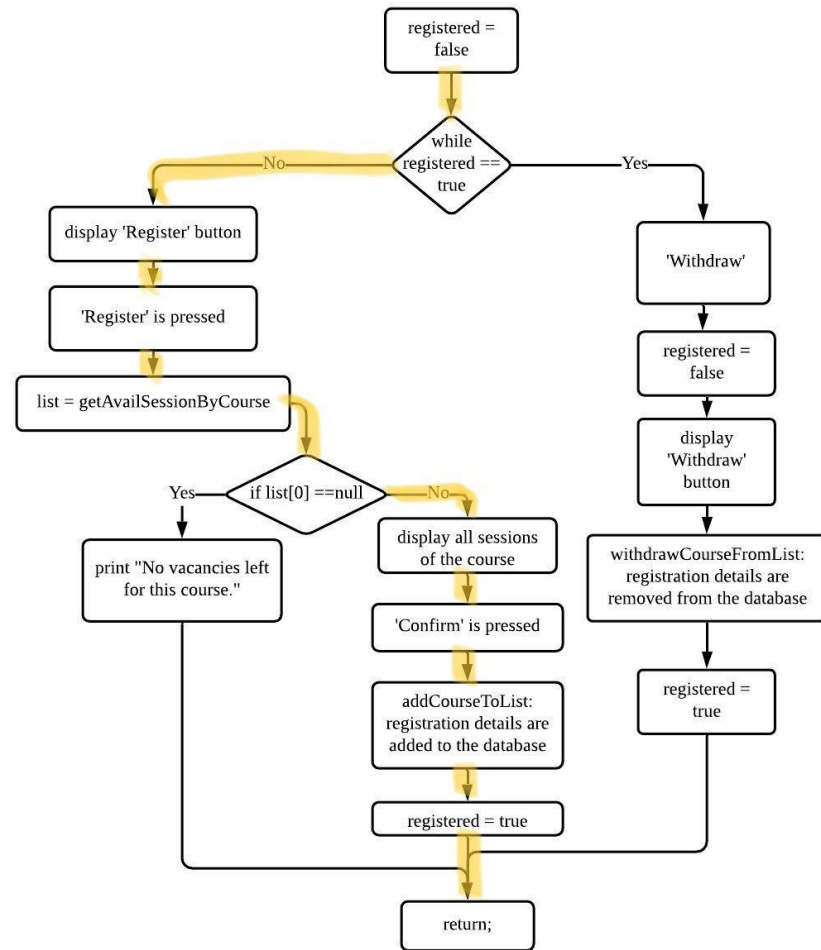
	Email Address	Password	Expected Output	Actual Output
Incomplete Fields	-empty-	-empty-	Homepage	"Email cannot be empty" "Password cannot be empty"
Valid User email , invalid password	MarjorieGallagher@gmail.com	1234	Homepage	"The email or password is invalid or the account does not exist"
Invalid User email, Valid password	something	Marjor1e@Gallagher	Homepage	"The email or password is invalid or the account does not exist"
Valid User email , valid password	MarjorieGallagher@gmail.com	Marjor1e@Gallagher	Homepage	Homepage



White Box Testing - Course Registration/Withdrawal

Path 1: Successful Registration

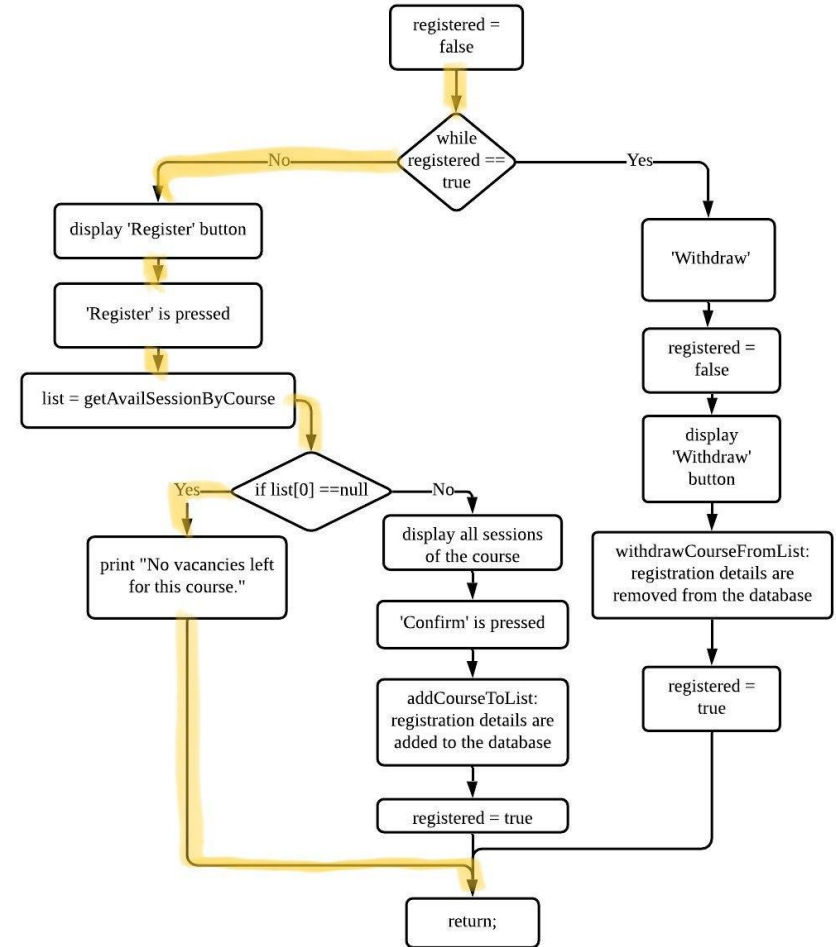
```
Run: main.dart x
Console
I/flutter ( 8146): Registered is false.
I/flutter ( 8146): Register button is displayed.
I/flutter ( 8146): Any available sessions retrieved via
'getAvailSessionByCourse'.
I/flutter ( 8146): Available sessions:
I/flutter ( 8146): 40
I/flutter ( 8146): 41
I/flutter ( 8146): 42
I/flutter ( 8146): There are available sessions.
I/flutter ( 8146): Available sessions are displayed.
I/flutter ( 8146): User confirms registration.
I/flutter ( 8146): 'addCourseToList' in UserController calls
DBHelper to add the registration details to the database.
I/flutter ( 8146): 'saveRegister' in DBHelper is used to save the
registration entry into the database.
I/flutter ( 8146): Registered is set to true.
I/flutter ( 8146): End of manual trace.
```



White Box Testing - Course Registration/Withdrawal

Path 2: No available sessions for the Course

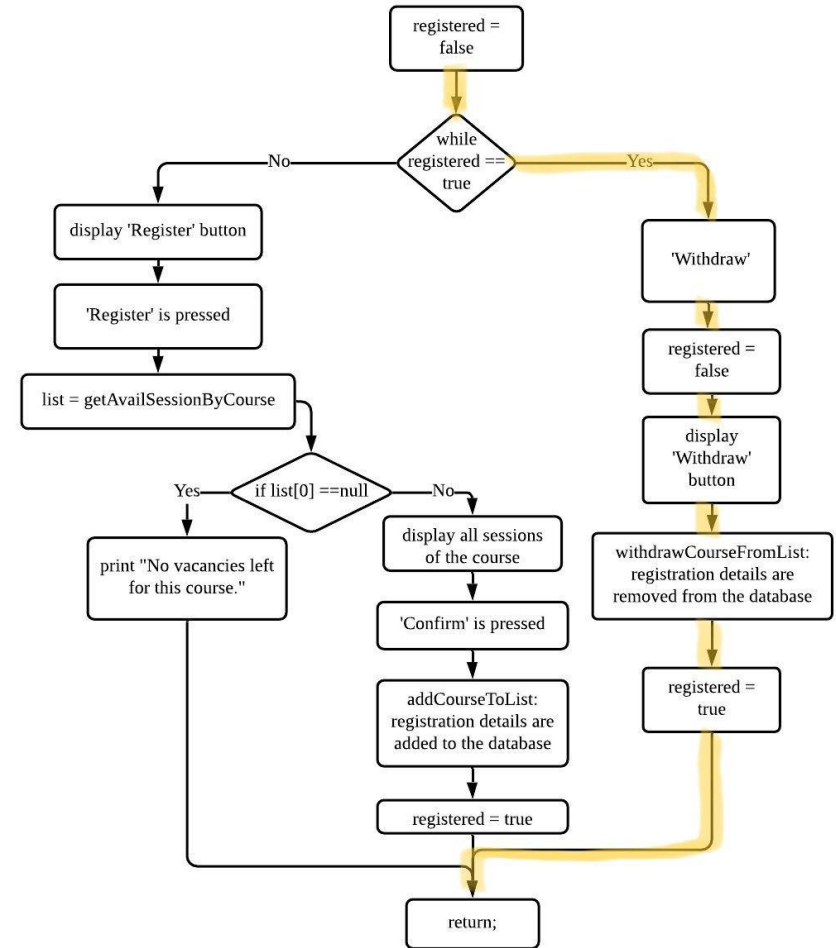
```
Run: main.dart
I/flutter ( 8146): Registered is false.
I/flutter ( 8146): Register button is displayed.
I/flutter ( 8146): No available sessions.
I/flutter ( 8146): Error message: 'There are no more vacancies for this course' is displayed.
I/flutter ( 8146): End of manual trace.
```



White Box Testing - Course Registration/Withdrawal

Path 3: Course already registered for -> Withdraw

```
Run: main.dart x
Console
I/flutter ( 8418): Registered is true.
I/flutter ( 8418): Withdraw button is displayed.
I/flutter ( 8418): 'withdrawCourseFromList' in UseController calls DBHelper to remove registration entry from database.
I/flutter ( 8418): 'deleteRegisterByUser' finds the correct registration entry and removes it from the database.
I/flutter ( 8418): End of manual trace.
```





04

FUTURE IMPROVEMENTS





4. FUTURE IMPROVEMENTS

- Map feature - User can check out the location of the courses in Google Maps.
- Reviews/chatbot - Establish a communication channel between user and vendor.
- Social media sharing - Allow vendors to promote their courses on other social media accounts, and allow user to share the courses to other users/non-users
- Linking up with schools - schools could partner up with a vendor to hold specialised lessons or workshops that go beyond the constraints of a classroom.





THANKS



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**