5. Results and Discussions

5.1 Test Results

5.1.1 Unit Tests

Starting from the bottom the first test level is "Unit Testing". It involves checking that each feature specified in the "Component Design" has been implemented in the component.

In theory an independent tester should do this, but in practice the developer usually does it, as they are the only people who understand how a component works. The problem with a component is that it performs only a small part of the functionality of a system, and it relies on co-operating with other parts of the system, which may not have been built yet. To overcome this, the developer either builds, or uses special software to trick the component into believe it is working in a fully functional system.

The summary of unit tests is provided below:

For User:

UNIT	PURPOSE	VERIFIED
Search Train	This unit search the trains.	Yes
Train details	This unit shows the trains of a particular source to destination in a particular date and a specific seat .	Yes
Book ticket.	This unit user can select a particular train and book ticket .	Yes
Login.	This unit login the registered user and create an account for a new user.	Yes
Fill the booking form.	User fill the form to book ticket.	Yes
payment.	User fill the form and pay the money with the help of credit card.	Yes
Ticket no	This unit allows to show his ticket no.	Yes
Cancel ticket	This unit allows user to cancel ticket.	Yes

For Administrator

UNIT	PURPOSE	VERIFIED
Administrator Login	This unit detects the authorization of the Administrator.	Yes
Change the train details	This unit allows the administrator to change the train details.	Yes

Integration Testing

As the components are constructed and tested they are then linked together to check if they work with each other. It is a fact that two components that have passed all their tests, when connected to each other produce one new component full of faults. These tests can be done by specialists, or by the developers.

Integration Testing is not focused on what the components are doing but on how they communicate with each other, as specified in the "System Design". The "System Design" defines relationships between components.

The tests are organized to check all the interfaces, until all the components have been built and interfaced to each other producing the whole system.

Thus this test was successfully done. No conflicts or inconsistencies were detected.

System Testing

Once the entire system has been built then it has to be tested against the "System Specification" to check if it delivers the features required. It is still developer focused, although specialist developers known as systems testers are normally employed to do it.

In essence System Testing is not about checking the individual parts of the design, but about checking the system as a whole. In fact it is one giant component.

System testing can involve a number of specialist types of test to see if all the functional and non - functional requirements have been met. In addition to functional requirements these may include the following types of testing for the

non - functional requirements:

- Performance - Are the performance criteria met?
- Volume - Can large volumes of information be handled?
- Stress - Can peak volumes of information be handled?
- Documentation - Is the documentation usable for the system?
- Robustness - Does the system remain stable under adverse circumstances?

The system was found to perform its function properly under all conditions.

Acceptance Testing

Acceptance Testing checks the system against the "Requirements". It is similar to systems testing in that the whole system is checked but the important difference is the change in focus:

Systems testing checks that the system that was specified has been delivered. Acceptance Testing checks that the system will deliver what was requested.

The customer should always do acceptance testing and not the developer. The customer knows what is required from the system to achieve value in the business and is the only person qualified to make that judgment. This testing is more of getting the answer for whether is the software delivered as defined by the customer. It's like getting a green flag from the customer that the software is up to the expectation and ready to be used.

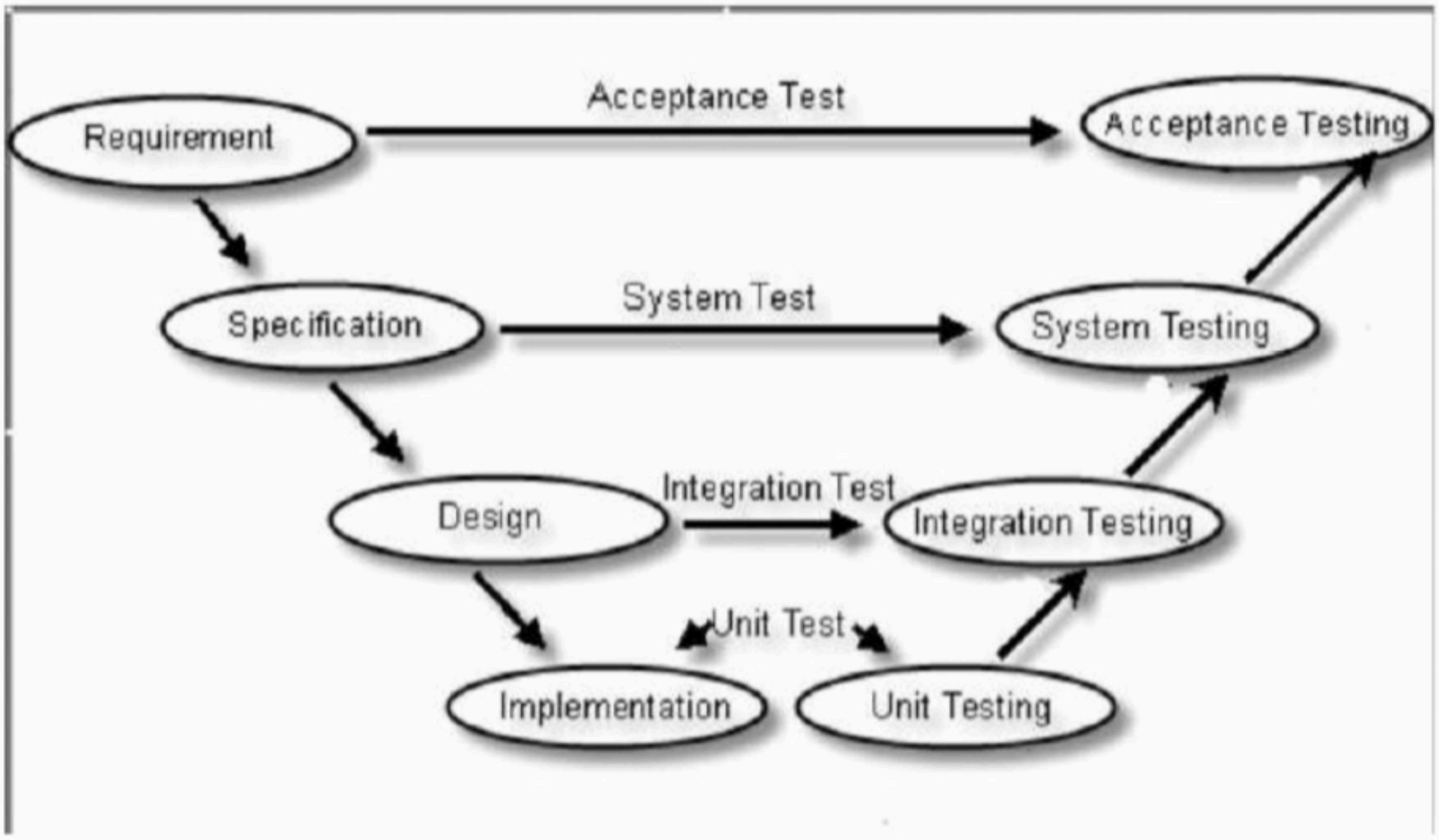

Figure 1.4: - V model cycle flow

5.1.2 Result

Online railway ticket reservation system was successfully designed and developed as per the specifications. It was extensively tested using a database which contains data similar to what can be expected in an actual database. The system was found to work satisfactorily without any errors under all conditions.