RAMAIAH UNIVERSITY
OF APPLIED SCIENCES

ASSIGNMENT

Course Code	CSC210A
Course Name	Software Development Fundamentals
Programme	B.Tech
Department	CSE
Faculty	FET

Name of the Student	Satyajit Ghana
Reg. No	17ETCS002159
Semester/Year	04/2019
Course Leader/s	Ms. Sahana P Shankar

Declaration Sheet

Student Name	Satyajit Ghana		
Reg. No	17ETCS002159		
Programme	B.Tech	Semester/Year	04/2019
Course Code	CSC210A		
Course Title	Software Development Fundamentals		
Course Date		to	
Course Leader	Ms. Sahana P Shankar		

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

List of Figures

Figure No. **Title of the figure** **Pg.No.**

Question No. 1

Solution to Question No. 1 Part A:

A 1.1 Introduction to the topic:

Quality of software products depends upon various phase of software development process. Process of software development is used to create and achieve quality in software products. Software development process uses four main phases which have its own importance for development. Software quality is a conformance to requirements which is divided into functional and non-functional requirements. The objective of this debate is to present a review on the impact of software development process on software quality. During analysis we observe that software architecture is more important phase than other phase because it provides abstract representation of overall structure of software.

Quality can refer to how maintainable software is, speed, accuracy, stability, usability, readability, testability, security, reliability, size, cost, and number of flaws or bugs as well as to less considerable qualities like conciseness, elegance, style, user satisfaction, and along with numerous additional attributes. How best to create high quality software product is a complex, separate, and contentious problem covering software design methods and principles, so called best practices for writing, developing, and testing code, with broader management issues such as process, finest team size, how best to deliver software on time and fast as possible, within budget, hiring practices, work place culture, and so forth. All this belong to software engineering.

A 1.2 Importance of requirements engineering in the early stages of software development:

Requirements analysis is important to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs and defined to a level of detail sufficient for system design. Software requirements include business requirements, user requirements, system requirements, external interface requirement, functional requirements, and non-functional requirements. In requirement statements every individual business, user, functional, and nonfunctional requirement would exhibit the qualities. Characteristics of requirement statements are complete, correct, feasible, necessary, prioritized, unambiguous, and verifiable. But it’s not enough to have excellent individual requirement statements. So, requirements collections are used to collect a set of requirement or group of requirements.

Requirements should state what to do and the design should describe how it does this. Characteristics of requirements are complete, consistent, modifiable, and traceable which reflect the software quality. Quality is heavily dependent on functional or non-functional requirements. In this section we have analyzed the various parameters of requirement i.e. consistency, completeness or correctness and modifiable. Consistency means providing predictable, maintainable and reliable results to the customer. Consistency refers to situations where a specification contains no internal contradictions, whereas completeness refers to situations where a specification entails everything that is known to be "true" in a certain context. It contains all necessary information to avoid ambiguity and need no amplification to enable proper implementation and verification.

Correctness is usually meant to be the combination of consistency and completeness. Correctness is often more pragmatically defined as satisfaction of certain business goals.

Success of software organization completely depends on user who use software and satisfy with software quality. Every software organization wants to develop quality product within given time and cost. To develop quality product first step should be a clear, complete, and precise software requirement. After requirement software design accordingly then software coding is start for executable version of software. Finally testing of software is started to check the software errors and bugs. All this belong to software quality.

(T. ur Rehman, M. N. Khan, and N. Riaz, 2013) (K. Wiegers and J. Beatty, 2013) (D. Zowghi and V. Gervasi)

A 1.3 Justification with stance taken and conclusion:

As discussed in A1.2 that Requirements Engineering is an important phase of the Software Development Life Cycle, but I do not agree that it's just the requirement engineering that affects the quality of the software. Software product evolves over their lifetime because evolution of software product is a continuous and natural process in which software must respond to developing and changing requirement, architecture, and environment.

Design patterns are reusable solutions to general design problems that are estimated to improve various quality attributes. Software programming languages are having important role to develop quality product. It is essential to choose programming language based on software requirement. Success of software product directly related to users who use the software. Usability of software product is the one of the most important quality attributes for user. Usability of software product can be achieved using software process.

Question No. 2

Solution to Question No. 1 Part B:

B 1.1 Introduction to the online reservation system:

The definition of online booking software eliminates the need to facilitate online bookings. In online booking system is software which allows a potential customer to book and pay for an activity or service directly through your website. That means from the moment a customer decides they want to book to choosing a date, picking a time and paying for the booking, everything is handled online, greatly reducing the workload on your staff and removing the opportunity for double-bookings.

Advanced systems like ours allow customers to book through a variety of methods online, including mobile, greatly expanding the potential for bookings for your business, and better leveraging an increasingly social internet.

Software management is made simple by providing an admin user that manages staff, the Staff users manage the users, and the bus routes, etc. Everything is automated, for example if there are changes to the bus schedule then it is intimated to the user via SMS, if there is a refund made by the user, the Payment Gateway reverses the transaction and send the amount back to the user account. The login and register are taken care with the Authentication Service. Splitting the work into these Actors makes it simpler and easier to manage.

B 1.2 Functional Requirements Specification:

FR1*:* The software should be able to allow the user to sign up if he/she is a new user else allow the user to log in with his/her email id or phone number along with a password.
Stakeholders owning the requirements: staff, new user
Dependent on requirements: --

FR2: The software should be able to authenticate a registered user, and reset password via OTP on registered mobile number.
Stakeholders owning the requirements: registered user
Dependent on Requirements: FR1

FR3: The software should allow the user to enter desired pickup and destination stops with date.

Stakeholders owning the requirements: registered user

Dependent on Requirements: FR1

FR4: The software should display the list of buses with its specifications like what amenities which bus provides at the desired route.

Stakeholders owning the requirements: staff, registered user

Dependent on Requirements: FR1, FR2

FR5: The software should allow the user to select a bus and book a seat in it.

Stakeholders owning the requirements: registered user

Dependent on Requirements: FR3

FR6: The software should allow the user to make payment for the tickets

Stakeholders owning the requirements: registered user

Dependent on Requirements: FR1, FR5

FR7: The software should allow the customer to apply discount codes at payment.

Stakeholders owning the requirements: registered user

Dependent on Requirements: FR1, FR6

FR8: The software should be able to display the confirmed tickets, containing the date, timing and seats details if the payment is successful.

Stakeholders owning the requirements: staff, registered user

Dependent on Requirements: FR1, FR6

FR9: The software should allow the staff to update changes (if there are any) in bus schedules. And inform the customers by sending a message on their registered mobile number.

Stakeholders owning the requirements: staff

Dependent on Requirements: --

FR10: The software should allow the user to cancel the tickets following which a refund should be initiated.

Stakeholders owning the requirements: registered user

Dependent on Requirements: FR1, FR6, FR8

FR11: The software must be able to authenticate staff users using their staff id and password

Stakeholders owning the requirements: staff, admin

Dependent on Requirements: --

FR12: The software must allow the staff users to create new bus routes and add additional buses.

Stakeholders owning the requirements: staff

Dependent on Requirements: FR11

FR13: The software must allow the admin to add staff users with their security clearance.

Stakeholders owning the requirements: staff, admin

Dependent on Requirements: --

B 1.3 Non-Functional Requirements Specification:

NFR1: The software should be user friendly.

NFR2: The software should have a modern, precise and accurate User Interface and User Experience.

NFR3: The software should have a safe and secure payment environment.

NFR4: The software should be efficient in terms of performance and response time.

NFR5: The software should be free from bugs and technical errors.

B 1.4 Conclusion

The process to gather the software requirements from client, analyze and document them is known as requirement engineering. The goal of requirement engineering is to develop and maintain sophisticated and descriptive 'System Requirements Specification' document. The SRS is developed based the agreement between customer and contractors.

It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of Software system. To achieve this we need to continuous communication with customers to gather all requirements.

Question No. 3

Solution to Question No. 2 Part B:

B 2.1 Introduction to structural design documentation:

Structural models of software display the organization of a system in terms of the components that make up that system and their relationships.

The aim of structured design is to transform the results of the structured analysis (i.e. a DFD representation) into a structure chart. Structured design provides two strategies to guide transformation of a DFD into a structure chart.

- Transform analysis
- Transaction analysis

Normally, one starts with the level 1 DFD, transforms it into module representation using either the transform or the transaction analysis and then proceeds towards the lower-level DFDs. At each level of transformation, it is important to first determine whether the transform or the transaction analysis is applicable to a particular DFD. These are discussed in the subsequent subsections.

B 2.2 User Characterization:

FR1: The software should be able to allow the user to sign up if he/she is a new user else allow the user to log in with his/her email id or phone number along with a password.

Use Case	Register
Actors	New user
Description	The user should be able to make a new account if he's not having one already

FR2: The software should be to able verify password allow the user to reset the password if he/she forgets it, by sending an OTP to the registered phone number.

Use Case	Reset password
Actors	Registered user, New user
Description	If the password entered by the user is incorrect, display an alert to reset the password if he has forgotten it.

FR3: The software should allow the user to enter desired pickup and destination stops with date.

Use Case	Enter pickup and destination
Actors	Registered user
Description	A passenger must able to enter pickup and drop locations.

FR4: The software should be able to display the list of buses available at the desired route on the specified date with all the specifications of the buses.

Use Case	Display list of buses
Actors	Registered user
Description	The software displays the list of buses available at the desired route on the specified date with all the specifications of the buses like what amenities which bus provides.

FR5: The software should be able to allow the user to select a bus and book a seat in it.

Use Case	Book seat
Actors	Registered user
Description	The user must be able to choose his desired number of seats.

FR6: The software should be able to allow the user to make payment.

Use Case	Make payment
Actors	Registered user
Description	The user must be able to pay for their tickets.

FR7: The software should be able to allow the customer to apply promo codes.

Use Case	Apply promo codes

Actors	Registered user
Description	The user must be able to apply promocodes to reduce the amount of ticket bill.

FR8: The software should be able to display the confirmed tickets, containing the date, timing and seats details if the payment is successful.

Use Case	Display confirmed tickets
Actors	Registered user, staff
Description	The user gets a softcopy of his/her ticket.

FR9: The software should be able to allow the staff to update changes (if there are any) in bus schedules.

Use Case	Update bus Schedule
Actors	Staff
Description	The staff users updates changes (if there are any) in bus schedules and informs the customers by sending a message on their registered mobile number.

FR10: The software should be able to allow the user to cancel the tickets following which a refund should be initiated.

Use Case	Cancellation and refund
Actors	Registered user, Staff
Description	The user cancels the ticket and save his/her money from getting wasted if there is some sudden change in his plan and he/she won't be able to board the bus at that particular time. The ticket amount is refunded to his/her bank account.

FR11: The software must be able to authenticate staff users using their staff id and password.

Use Case	Staff login
Actors	Registered user, Staff

Description	The staff logs in to the software with his/her staff id and login credentials

FR12: The software must allow the staff users to create new bus routes and add additional buses.

Use Case	Add Bus Route
Actors	Staff
Description	The Staff adds additional Bus Routes if required to fulfil the ticket demands.

FR13: The software must allow the admin to add staff users with their security clearance.

Use Case	Add Staff User
Actors	Admin
Description	The Admin adds staff users with different security clearance, such as a staff that can manage buses and routes, a staff that handles accounting, and a staff to manage refunds and cancellation.

B 2.3 Use-case diagram with Use-case specification:

Use-Case Specification:

Use-Case	Register
Description	Allows a new user to register
Pre-Condition	The user must not be already registered
Basic Path	The New user clicks on Register and enter correct details along with verified mobile number upon which he/she is registered.

Use-Case	Login
Description	Allows a registered user to login
Pre-Condition	The user must be registered

Basic Path	The Registered user enters valid credentials, which is authenticated by Authentication Service upon which he/she can enter the software.

Use-Case	Reset Password
Description	Allows a registered user to reset his/her password
Pre-Condition	The user must be registered
Basic Path	The New user clicks on Forgot Password and enters the registered mobile number, using the correct OTP he/she resets the password for his/her account.

Use-Case	Staff Login
Description	Allows a Staff user to Login
Pre-Condition	The staff must be valid
Basic Path	The staff user enters his/her valid staff id and credentials which is verified by Authentication Service, and correct access level is given to the staff.

Use-Case	Display Bus List
Description	Displays the list of buses for the given specifications
Pre-Condition	The User must be registered and should have selected the pickup and destination address.
Basic Path	The software displays the list of buses to the user.

Use-Case	Displayed Confirmed Tickets
Description	Allows the registered users to display his/her booked tickets.
Pre-Condition	The user must be registered and should have made successful payment for the ticket
Basic Path	The User goes to his/her booking history and views the previously booked tickets.

Use-Case	Ticket Cancellation
Description	Allows the user to cancel his/her booked ticket

Pre-Condition	The user must have successfully made payment for the booked ticket
Basic Path	The user selects cancel ticket on the list of previous bookings, and the amount is refunded to his/her bank account.

Use-Case	Make Payment
Description	Allows the user to make payment for the ticket
Pre-Condition	The user must have selected the desired bus from the list of buses.
Basic Path	The user select payment method and pays using his/her credit, debit card, which is processed by the Payment Gateway.

Use-Case	Add Staff User
Description	Allows the admin to create staff users
Pre-Condition	None
Basic Path	The Admin creates more staff members with appropriate security clearance and staff id with appropriate credentials.

Use-Case	Update Bus Schedule
Description	Allows staff users to alter the bus schedule
Pre-Condition	The logged in user must be a staff
Basic Path	The Staff user changes the bus schedule which updates the schedule and informs the customers about the change via SMS.

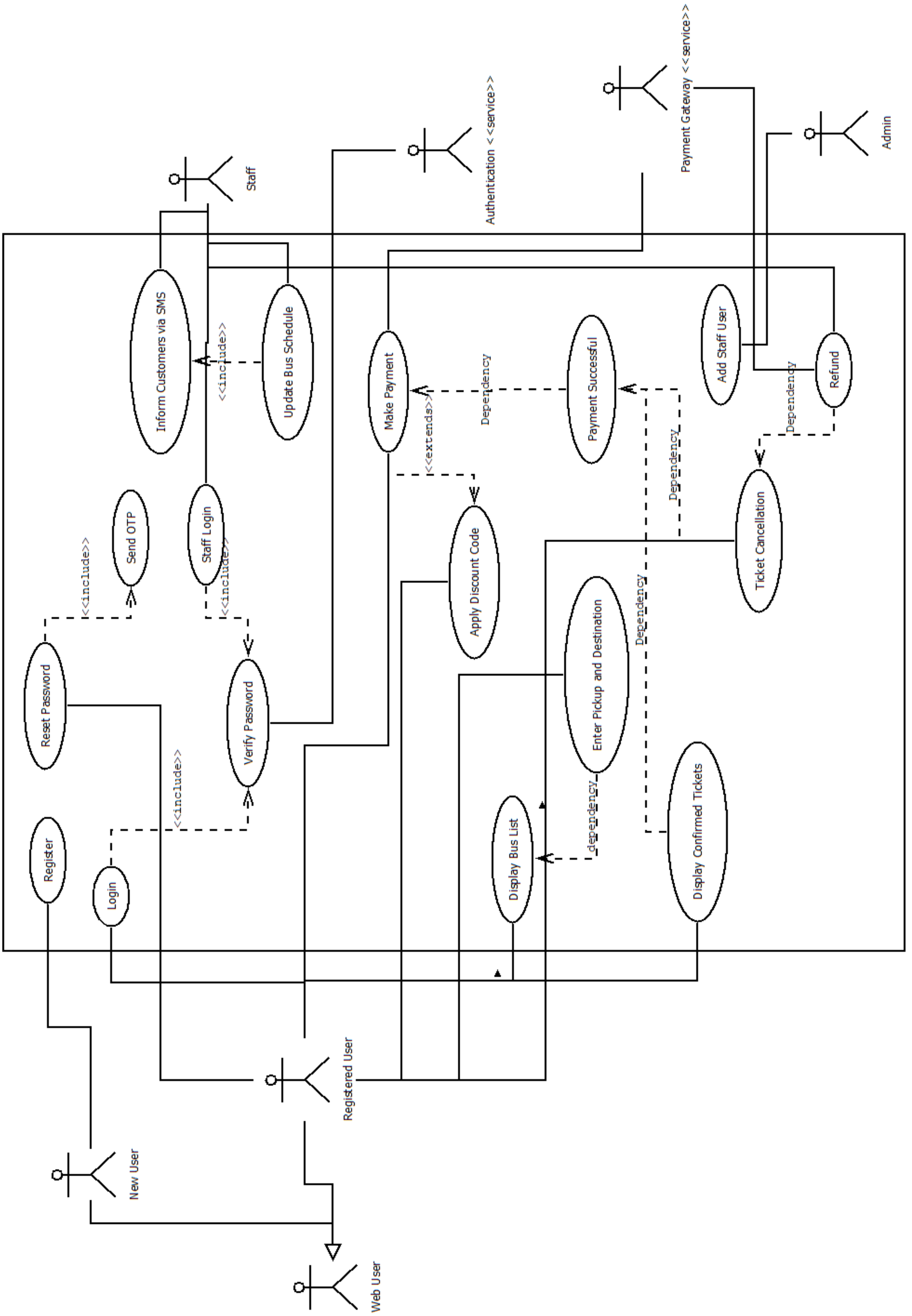

Figure 1 Use-Case Diagram

B 2.4 Conclusion:

A Structured Design for the given problem of Bus Ticket Reservation System is made by specifying the Functional Requirements, Non-Functional Requirements and the then performing Use-Case Scenario and making the Use-Case diagram and the Use-Case Specifications. Doing this gives a more rigid idea about the software specifications and makes the stand points clear during the development phase of the software.

Structural patterns ease the design of a program by identifying a way to realize relationships between entities. The difference between class patterns and object patterns is that class patterns describe how inheritance can be used to provide more useful program interfaces. Object patterns, on the other hand, describe how objects can be composed into larger structures using object composition, or the inclusion of objects within other objects.

"Structural class patterns use inheritance to compose interfaces or implementations."

[GoF, "Design Patterns", Addison Wesley, ISBN 0201633612]

Bibliography

1. Basic Concepts in Object Orientation, Structured Design, IIT Kharagpur, Module 6, Lecture 13
2. https://www.softwaretestingclass.com/software-requirement-specification-srs/
3. D. Zowghi and V. Gervasi, “The Three Cs of requirements: consistency, completeness, and correctness,” In International Workshop on Requirements Engineering: Foundations for Software Quality, pp. 155-164, Sep 2002
4. K. Wiegers and J. Beatty, Software requirements, Pearson Education, 2013.
5. T. ur Rehman, M. N. Khan, and N. Riaz, “Analysis of requirement engineering processes, tools/techniques and methodologies,” International Journal of Information Technology and Computer Science (IJITCS), Vol. 5, Issue 3, pp. 40-48, Feb 2013. doi: 10.5815/ijitcs.2013.03.05.