

**Bachelor of Technology
In
Computer Science And Engineering**

*Synopsis of
Software Requirement Specification (SRS)
of
"Paytm "*

Name : Rajdeep Roy Chowdhury
Reg No. : 11712546
Section : K17SJ
Roll No. : K17SJA14
Group : 1
Session : 2018-2019

**Lovely Professional University
Punjab, India**

Table of Contents

- 1. Introduction..... 1
- 2. Specific Requirements..... 2
 - a. Functional Requirements.....2-6
 - b. Non-functional Requirements..... 7

1.Introduction

Paytm stands for Pay through mobile and focuses on the usage of e-cash. This is a synopsis of the SRS of Paytm which aims to deliver all the functional and nonfunctional requirements to anyone working in the Paytm team.

1.1. Why Paytm?

Not everyone can carry cash everywhere. Moreover there is no change available all the time. Even the ATMs may be out-of-cash sometimes. To resolve some of the issues with traditional cash-based system, Paytm like systems are borned.

What makes paytm so special? The availability of tons of services make it stand out of it competitors. Moreover it's UI is super reliable and easy to use.

1.2. What's inside the SRS Synopsis?

This document consists of Functional and Non-functional Requirements of Paytm.

1.3. Disclaimer:

The Synopsis of the SRS of Paytm is just a case-based study allotted to me at Lovely Professional University as a part of the curriculum of B. Tech CSE (3rd Semester). This has nothing to do with the Original Paytm Application.

2. Specific Requirements

2.1. Functional Requirements

2.1.1. Functional Requirement 1: SIGNUP

Input: User would enter all the required details such as Mobile Number, Name, DOB, password, etc

Process: All the data would be sent to the server for registration. Verification via OTP or mail would be done.

Output: User would get to know whether account creation was successful or not.

2.1.2. Functional Requirement 2: LOGIN

Input: User would input his/her credentials to the login screen and press the login button

Process: Credentials would be sent to the servers for validation and processing

Output: On success user would be able to see Home screen

2.1.3. Functional Requirement 3: Choose Language

Input: Combo box input would be present to select from a number of native languages such as

Bengali, Hindi, English, etc where the default would be set to English.

Process: On changing the selection, all the contents would be translated to the chosen language.

Output: User would be able to view each and every component of Paytm in their native language.

2.1.4. Functional Requirement 4: 24x7 Help

Input: User would tap on the option and type his/her query

Process: User's query would be sent to the Customer care. If it would be present in the FAQ then he/she would be redirected to the Community Forums activity.

Output: User would be provided the solution instantly.

2.1.5. Functional Requirement 5: Community Forum

Input: User would enter the Community Forum and can view all the Frequently asked by the other users. A search bar would be present for the user to type.

Process: The query would be processed using Natural Language processing and the possible solution forum would be shown. If a frequent number of the same query would be received then the solution would be added to the FAQ list. Prioritizing of the Questions would be also done.

Output: User would be able to view all the possible solutions to his query.

2.1.6. Functional Requirement 6: Pay Money to Mobile Number

Input: User would be entering the payee's phone number and the amount

Process: Entered amount would be deducted from the payer's account and would be credited to the payee's account.

Output: On successful transaction, success message would be shown, else transaction failed would be shown and be notified in the notification section.

2.1.7. Functional Requirement 7: Pay Money to Bank account number

Input: User would be entering the bank account number of the payee and the amount.

Process: Deducting a little amount service charge, money would be transferred to the payee's Bank Account.

Output: On successful transaction, success message would be shown, else transaction failed would be shown and be notified in the notification section.

2.1.8. Functional Requirement 8: Pay Money through QR Code

Input: User would scan the QR code of the Payee through the camera of the phone and then enter the money.

Process: Money would be transferred to the payee's paytm wallet instantly.

Output: On successful transaction, success message would be shown, else transaction failed would be shown and be notified in the notification section.

2.1.9. Functional Requirement :9 View Paytm passbook

Input: User would tap on the View Paytm Passbook option.

Process: All the transaction details would be fetched from the paytm servers.

Output: The fetched data would be shown in a tabular format to the user.

2.1.10. Functional Requirement 10: Add money to paytm wallet

Input: Enter amount, select saved bank account or enter new bank account, enter the OTP.

Process: Money from the bank account would be credited to the paytm wallet.

Output: On success, money would be added to the paytm wallet and a toast message would be shown to the user.

2.1.11. Functional Requirement 11: Nearby KYC Point

Input: Enter location manually or get location manually using google location services.

Process: Current location of the user would be known and nearest KYC point would be searched.

Output: Nearest KYC Point's address, phone number would be shown. In addition to that, realtime map would be also provided.

2.1.12. Functional Requirement 12: Mobile Recharge

Input: Enter phone number to be recharged, amount, and check if any available offers.

Process: Selected amount would be deducted from the paytm wallet and paid to the service provider and thus phone number would be recharged.

Output: Recharge successful would be shown if successful.

2.1.13. Functional Requirement 13: KYC (Know Your Customer)

Input: User would input his/her aadhar card number.

Process: The aadhar card would be stored in the database of the paytm, and would be enqueued for verification through the paytm KYC point.

Output: On success, user would be shown a success message.

2.1.14. Functional Requirement 14: Security Settings

Input: User would be have three option to secure their paytm accounts. Change Password, Manage App lock, Set Passcode.

Process: App settings as well as remote account setting are updated.

Output: User's valuable money is more secured than ever.

2.1.15. Functional Requirement 15: Profile Settings

Input: Input fields for first name, last name, display name, mobile number, and display picture would be there. After all that, a save button would be also there.

Process: User details would be sent and updated in the paytm servers.

Output: On success, a message would be displayed to the user.

2.1.16. Functional Requirement 16: Offers

Input: User would enter the offers section.

Process: Current activity would be changed to the offers page.

Output: User would be able to see best deals, coupon codes, free vouchers in paytm.

2.1.17. Functional Requirement 17: Electricity Bill payments

Input: Input all the required fields in the screen such as Customer ID, Branch, etc and set amount to be paid.

Process: If entered amount exceeds the paytm wallet money then prompt the user to add more money. Else carry out the payment transaction and deduct the adjusted amount from the paytm wallet and send it to the Electricity Department.

Output: The electricity bill would be paid and notified to the user.

2.1.18. Functional Requirement 18: Movie Tickets

Input: user would enter number of people, venue, time, etc. He/she would tap on the pay button and wait for the transaction to be completed.

Process: Paytm would book the seats in favour of the user.

Output: On success, user would receive a confirmation message.

2.1.19. Functional Requirement 19: DTH Recharge

Input: User would enter the Viewing Card Number, active plan, and amount.

Process: Selected amount would be deducted from the paytm wallet and transferred to the DTH account.

Output: On success, DTH account would be recharged and notified.

2.1.20. Functional Requirement 20: Train tickets

Input: User would enter the source and destination and select the seater class and seat. He/she can also check availability of seats as well as the status of reservation in the same activity.

Process: Paytm would communicate with the APIs of the IRCTC and book the seats for the users. Train fare would be deducted from the paytm wallet, and ticket would be sent to the user's email.

Output: User would get confirmation mail from paytm as well as IRCTC on successful transaction.

2.1.21. Functional Requirement 21: School/College/University fees payment

Input: User would enter his/her Registration number and the basic details required. And payment would be verified through otp or verification call.

Process: Fees would be paid to the institution. And wallet money would be also deducted to balance the amount.

Output: User would be notified by the Institution and also by paytm message that his/her payment/transaction is successful.

2.1.22. Functional Requirement 22: Paytm Shopping

Input: User would enter the Paytm shopping option and browse through any category of items. User would select any item, apply coupon code, enter delivery address and contact number and place the order, that would be verified by OTP.

Process: Purchase request would be sent to the paytm servers and the item would be delivered to user. User would be also notified at each checkpoint of the delivery.

Output: User would be able to Purchase items from their e-wallet app.

2.1.23. Functional Requirement 23: Notifications

Input: User would enter the notifications section.

Process: User notifications would be push to the local device from the cloud servers.

Output: User would be updated with latest trends, transaction messages-- Success or failure both.

2.2. Non-functional Requirements

2.2.1. Non-functional Requirement 1: Usability

Prioritize the important functions of the system based on usage patterns. Frequently used functions should be tested for usability, as should complex and critical functions.

2.2.2. Non-functional Requirement 2: Reliability:

Users have to trust the system, even after using it for a long time. Create a requirement that data created in the system will be retained for a number of years without the data being changed by the system. It's a good idea to also include requirements that make it easier to monitor system performance.

2.2.3. Non-functional Requirement 3: Performance:

During the diwali sale, there would be maximum number of user connected to paytm. All the request should be handled properly and efficiently. Even if the request-load exceeds the limit, Software should be always ready for horizontal scaling of servers.

2.2.4. Non-functional Requirement 4: Supportability:

The system needs to be cost-effective to maintain. Maintainability requirements may cover diverse levels of documentation, such as system documentation, as well as test documentation, e.g. which test cases and test plans will accompany the system.

2.2.5. Non-functional Requirement 5: Security:

Since the user is providing his/her Banking Details, and other private documents, it is the responsibility of the Paytm Team to secure the data hackers or being leaked. Not only on the server side, client side should be protected too. Measures should be taken on both the sides.

2.2.6. Non-functional Requirement 6: Cross platform support:

Not all the users use android, and not all the users use iOS. Paytm must be supported in Multiple platforms with the same user experience. Basic operating system supported must be Android, iOS, Windows, Fire OS, and Tizen.