**

***Software Engineering***

***Software Requirements Specification***

***(SRS) Document***

*Ayush Pandey 42*

*Vaishnav Patil 53*

*Aditya Prajapati 55*

*Dhwani Prajapati 57*  
*SoftEng06*  
*www.radford.edu/~softeng06*  
*08/02/2023*

|  |
| --- |
| **Revisions** |

| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| Version 2.100  Version 2.118  Version 2.121  Version 2.130 | Team  Team  Team  Team | All sections being Filled  All sections being Filled  All sections being Filled  All sections being Filled | 07/11/19  3/11/20  14/03/21  18/11/21 |

|  |
| --- |
| **Review & Approval** |

**Requirements Document Approval History**

| **Approving Party** | **Version Approved** | **Signature** | **Date** |
| --- | --- | --- | --- |
| **Barry Chapman** | 2.130 |  | 15-02-22 |
| **Dr. T. L. Lewis** | 2.130 |  | 15-02-22 |

**Requirements Document Review History**

| **Reviewer** | **Version Reviewed** | **Signature** | **Date** |
| --- | --- | --- | --- |
| **Drashti Shrimal** | 2.130 |  | 22-02-22 |
| **Aniket Mishra** | 2.130 |  | 22-02-22 |

|  |
| --- |
| **Contents** |

[1. Introduction 3](#_gjdgxs)

[2. General Description 3](#_30j0zll)

[3. Functional Requirements 4](#_1fob9te)

[4. Interface Requirements 5](#_3znysh7)

[4.1 User Interfaces 5](#_2et92p0)

[4.2 Hardware Interfaces 5](#_tyjcwt)

[4.3 Communications Interfaces 5](#_3dy6vkm)

[4.4 Software Interfaces 5](#_1t3h5sf)

[5. Performance Requirements 6](#_4d34og8)

[6. Other non-functional attributes 6](#_2s8eyo1)

[6.1 Security 6](#_17dp8vu)

[6.2 Availibilty 7](#_3rdcrjn)

[6.3 Reliability 7](#_26in1rg)

[6.4 Maintainability 7](#_lnxbz9)

[6.5 Compatibility 7](#_35nkun2)

**1. Introduction**

**Introduction**

Gpay which stands for Google pay will be an online money transaction application. It’s main purpose to help people to transfer money sitting at home. All the service is free of cost, even transfer money with Zero fees straight from your account. Google pay also offers for recharge, bill payments, shopping and many.

**Scope of this Document**

After demonetization in India in November 2016 digital transaction became the main source of transfer money but before it, people were afraid of it as it’s a fraud. At the same time Jio launched internet connectivity at a very low cost so, digital transaction came in demand. As google pay transaction is straight from Bank account. Now, it is one of the largest mobile payment service platforms with the facility of one tap payment.

**Overview**

In this document, detailed design of the system with user interfaces will be described, Decomposition of the system with module decomposition, concurrent process decomposition and data decomposition, User Interface description, detailed design of the system is provided.

**Business Context**

Google Pay can be used for Business to get customer payments more quickly, more simply, and more securely in your business or online. Anyone with a universal payment identification code (UPIC) can use Google Pay to pay you for goods and services.

Example:

* Revenue Generation: Google Pay will generates revenue through various means, including transaction fees, interest on loans, and through partnerships with merchants and financial institutions.
* Data Monetization: Google Pay collects a vast amount of data on its users' financial transactions, which it can use to offer personalized financial services and monetize through targeted advertising and other initiatives.

## 2. General Description

**2.1 Product Functions**

Each Gpay account will have its own data in database to maintain its process and transaction even the users on the application. When we login to Gpay using Google PIN, it communicates with the database. Now we can pay to or request money from other Users. It will check your details with the bank and in the case of satisfaction your payment will be done. In case of order or shopping you will be redirected to the interface and again your details will be matched with the database.

**2.2** **Similar System Information**

Google Pay is an tightly integrated with other Google services, such as Gmail and Google Maps, which provides a seamless user experience for users who are already familiar with these services. It is available on Android and iOS, which provides a larger user base and wider platform reach compared to other digital payment platforms that are limited to specific operating systems. It employs advanced security features, such as tokenization and encryption, to protect users' financial information and transactions.

**2.3 User Characteristics**

There are several users of the Gpay

• Customer : Cutomer Interact with the Gpay via Application.It is very easy to use because it support one tap payment.

• Maintainer : Using Gpay is easy but to maintain, it needs effort. So, the programmers and coders work together to maintain the database and internal function to make the Gpay easy to use.

**2.4 User Problem Statement**

The need is to protect users' financial information and transactions.

**2.5 User Objectives**

To have access to insights and reports on their spending and transactions, the goal is to help users make more informed financial decisions.

**2.6 General Constraints**

Since your payment through Google Pay is directly tied to the smartphone and mobile app, you’ll be stuck if you are having trouble with your phone or mobile network.Moreover, Google Pay is only supported by a limited number of banks, so if your bank doesn't use Google Pay, then you can't accept it as a payment method. Similarly, if your customers' banks don't support Google Pay, they can't use it to make payments at your store. This is a downside for both consumers and businesses.

## 3. Functional Requirements

Functional Requiremnt 1

* Description: Login in Gpay(Check whether account exists or not)
* Input: Enter the details like mobile number registered in your bank account to communicate with the bank.
* Processing: Checking whether this account exists or not.
* Output: If account exists then user interface will be opened else Account doesn’t exist, You have to sign in to create a Gpay Account
* Sign in: To sign in in google pay you need your Email Id and name, your registered mobile number in bank.
* Security: Choose Screen lock or Google PIN to secure the application
* Add Bank Account: Choose the bank you want to use for the application.
* Processing: It will send a message to the bank by your registered Mobile number that whether this number is registered in your bank or not in case of satisfaction verification will be done.
* UPI PIN: Choose UPI PIN to secure the transaction which is of atmost 6 numbers. After this your bank account will be added to Gpay to use, and this UPI PIN will be asked each time when you will transfer money.

Functional Requirement 2

* Description: Pay or request money
* Input1:Select New section and Enter the Gpay registered mobile number whom you want to pay or you can directly send to the bank by entering Account number, Account holder name and IFSC code.
* Input2: Enter the Amount you want to pay Processing: After entering UPI PIN It checks whether your Account has required amount or not.
* Output: If no error will occur then payment will be successful, otherwise payment unsuccessful
* Error: Server error, Amount problem, internet Connectivity.

## 4. Interface Requirements

**4.1 User Interfaces**

**4.1.1 GUI**  
On Android devices, the Google Pay app features a bottom navigation bar with four main sections: Home, Payment, Passes, and Activity. The Home screen shows a list of recent transactions and offers, while the Payment section allows users to make payments and manage their payment methods. The Passes section displays loyalty cards, tickets, and other types of digital passes, and the Activity section shows a history of transactions and account activity.

On iOS devices, the Google Pay app has a similar design, but with a tab bar at the bottom instead of a navigation bar. The tabs include Pay, Explore, Passes, and Insights, and the app provides a similar set of features for managing payments, passes, and account activity.

**4.1.2 CLI**  
Google Pay is primarily designed as a mobile app for use on Android and iOS devices, and it does not have a command line interface (CLI) that can be accessed directly from a command line terminal on a computer or server.

**4.1.3 API**  
The Google Pay API provides a set of REST APIs that enable developers to process payments, manage payment methods, and integrate loyalty programs and other features into their own applications. The API supports a variety of payment methods, including credit and debit cards, digital wallets, and bank transfers.

**4.2 Hardware Interfaces**

* Near Field Communication (NFC): Google Pay uses NFC technology to enable contactless payments at merchants that support it.
* Secure Element (SE): Google Pay also requires a secure element to store payment credentials and ensure the security of payment transactions.
* Biometric authentication: To ensure the security of payment transactions, Google Pay also relies on biometric authentication features like fingerprint scanning or facial recognition.
* Internet connection: Google Pay also requires an internet connection to function properly, as it needs to communicate with payment networks and other services to process payments and manage payment information.

**4.3 Communications Interfaces**

* Internet connectivity: Google Pay requires a reliable internet connection to function properly. This can be a Wi-Fi connection or a cellular data connection, depending on the device and the user's preference.
* Payment network and processor interfaces: Google Pay communicates with payment networks and payment processors to process payment transactions.
* Merchant interfaces: Google Pay also needs to communicate with merchant systems to authorize payment transactions and update account information.
* Operating system interfaces: Google Pay interfaces with the underlying operating system of the mobile device to access hardware features like the camera, NFC, biometric sensors, and other components.

**4.4 Software Interfaces**

* Payment gateway interfaces: Google Pay communicates with payment gateways to process payment transactions.
* Mobile operating system interfaces: Google Pay interfaces with the underlying operating system of the mobile device to access hardware features like the camera, NFC, biometric sensors, and other components.
* Cloud-based services interfaces: Google Pay relies on cloud-based services to store and manage payment credentials, loyalty program information, and other data.
* Third-party service interfaces: Google Pay may also interface with third-party services to offer additional features, such as loyalty programs or special offers.

## 5. Performance Requirements

* Fast transaction processing: Google Pay must process payment transactions quickly and reliably, with minimal latency or delay, to ensure a seamless user experience.
* High availability: Google Pay must be highly available, with minimal downtime or service interruptions. This requires robust infrastructure and redundancy measures to ensure that the app is always available for users.
* Scalability: Google Pay must be designed to scale to accommodate increasing volumes of users and transactions, without compromising on performance or security. This requires a flexible and scalable architecture that can adapt to changing demand.

## 6. Other non-functional attributes

### 6.1 Security

Google Pay must be designed with strong security measures to protect user data and prevent fraud. This requires adherence to industry standards like PCI DSS, strong encryption, and two-factor authentication, as well as regular security audits and vulnerability testing. Google Pay uses strong encryption to protect user data, including AES-256 and RSA encryption algorithms. User data is encrypted both in transit and at rest, and encryption keys are stored securely in Google's data centers.

### 6.2 Availability

Google Pay must be highly available, with minimal downtime or service interruptions. This requires robust infrastructure and redundancy measures to ensure that the app is always available for users.

### 6.3 Reliability

The reliability of Google Pay depends on the underlying infrastructure, including servers, network components, and data centers.Google Pay undergoes rigorous testing to ensure that it is functioning properly and free of bugs or errors. This includes unit testing, integration testing, and user acceptance testing, as well as regular load testing to ensure that the app can handle high volumes of traffic.

### 6.4 Maintainability

Google Pay must be maintainable, with a modular and extensible architecture that enables easy updates and bug fixes. The code must be well-organized and documented, with clear separation of concerns and minimal dependencies.

### 6.5 Compatibility

### Android devices: Google Pay is designed for Android devices running Android 5.0 or later. The app is available on the Google Play Store and can be installed on a wide range of Android smartphones and tablets.

### iOS devices: Google Pay is also available on iOS devices running iOS 12.0 or later. The app can be downloaded from the App Store and installed on iPhones and iPads.