***EXPERIMENT 1***

**Aim**: Understanding an SRS.

### Requirements:

**Software Requirements:** Star UML, Windows XP, **Theory:**

An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies *at a particular point in time* (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an eCommerce Web site, and so on) must provide, as well as states any required constraints by which the system must abide. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

It's important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

1. well-designed, well-written SRS accomplishes four major goals:
   * It provides feedback to the customer. An SRS is the customer's assurance that the development organization understands the issues or problems to be solved and the software behavior necessary to address those problems. Therefore, the

SRS should be written in natural language (versus a formal language, explained later in this article), in an unambiguous manner that may also include charts, tables, data flow diagrams, decision tables, and so on.

* + It decomposes the problem into component parts. The simple act of writing down software requirements in a well-designed format organizes information, places borders around the problem, solidifies ideas, and helps break down the problem into its component parts in an orderly fashion.
  + It serves as an input to the design specification. As mentioned previously, the SRS serves as the parent document to subsequent documents, such as the software design specification and statement of work. Therefore, the SRS must contain sufficient detail in the functional system requirements so that a design solution can be devised.
  + It serves as a product validation check. The SRS also serves as the parent document for testing and validation strategies that will be applied to the requirements for verification.

SRSs are typically developed during the first stages of "Requirements Development," which is the initial product development phase in which information is gathered about what requirements are needed--and not. This information- gathering stage can include onsite visits, questionnaires, surveys, interviews, and perhaps a return-on-investment (ROI) analysis or needs analysis of the customer or client's current business environment. The actual specification, then, is written after the requirements have been gathered and analyzed.

# SRS should address the following

The basic issues that the SRS shall address are the following:

1. ***Functionality****.*What is the software supposed to do?
2. ***External interfaces****.*How does the software interact with people, the system’shardware, other hardware, and other software?
3. ***Performance****.*What is the speed, availability, response time, recovery time ofvarious software functions, etc.?
4. ***Attributes****.*What are the portability, correctness, maintainability, security, etc.considerations?
5. ***Design constraints imposed on an implementation****.*Are there any requiredstandards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.

# Characteristics of a good SRS

An SRS should be

1. Correct

Unambiguous

1. Complete
2. Consistent
3. Ranked for importance and/or stability
4. Verifiable
5. Modifiable
6. Traceable

**Correct** - This is like motherhood and apple pie. Of course you want the specification tobe correct. No one writes a specification that they know is incorrect. We like to say - "Correct and Ever Correcting." The discipline is keeping the specification up to date when you find things that are not correct.

**Unambiguous -** An SRS is unambiguous if, and only if, every requirement stated thereinhas only one interpretation. Again, easier said than done. Spending time on this area prior to releasing the SRS can be a waste of time. But as you find ambiguities - fix them.

**Complete -** A simple judge of this is that is should be all that is needed by the softwaredesigners to create the software.

**Consistent -** The SRS should be consistent within itself and consistent to its referencedocuments. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another.

**Ranked for Importance -** Very often a new system has requirements that are reallymarketing wish lists. Some may not be achievable. It is useful provide this information in the SRS.

**Verifiable -** Don't put in requirements like - "It should provide the user a fast response."Another of my favorites is - "The system should never crash." Instead, provide a quantitative requirement like: "Every key stroke should provide a user response within 100 milliseconds."

**Modifiable -** Having the same requirement in more than one place may not be wrong - but tends to make the document not maintainable.

**Traceable -** Often, this is not important in a non-politicized environment. However, inmost organizations, it is sometimes useful to connect the requirements in the SRS to a higher level document. Why do we need this requirement?

# A sample of basic SRS Outline

* 1. **Introdu**

**ction** 1.1 Purpose

* + 1. Document conventions
    2. Intended audience
    3. Additional information
    4. Contact information/SRS team members
    5. References

### Overall Description

* 1. Product perspective
  2. Product functions
  3. User classes and characteristics
  4. Operating environment
  5. User environment
  6. Design/implementation constraints
  7. Assumptions and dependencies
  8. **External Interface Requirements** 3.1 User interfaces
  9. Hardware interfaces
  10. Software interfaces
  11. Communication protocols and interfaces

### System Features

* 1. System feature
     1. Description and priority
     2. Action/result
     3. Functional requirements
  2. System feature B
  3. **Other Nonfunctional Requirements** 5.1 Performance requirements
  4. Safety requirements
  5. Security requirements
  6. Software quality attributes
  7. Project documentation
  8. User documentation

### Other Requirements

Appendix A: Terminology/Glossary/Definitions list Appendix B: To be determined

**Conclusion:** The SRS was made successfully by following the steps described above.

**SOFTWARE REQUIREMENTS SPECIFICATION**

**Pantry2Plate**

**Table of Contents**

## Introduction

* 1. *Purpose*
  2. *Scope*
  3. *Definitions, Acronyms, and Abbreviations*
  4. *References*
  5. *Overview*

## The Overall Description

* 1. *Product Perspective*
  2. *Product Functions*
  3. *User Characteristics*
  4. *Constraints*
  5. *Assumptions and Dependencies*

## External Interface Requirements

* 1. *User Interfaces*
  2. *Hardware Interfaces*
  3. *Software Interfaces*
  4. *Communications Interfaces*

### System Features

1. **Other Non-Functional Requirements** 5.1 Performance Requirements
   * 1. Capacity
     2. Dynamic Requirements
     3. Quality
   1. *Software System Attributes*
      1. Reliability
      2. Availability
      3. Security
      4. Maintainability
   2. Business Rules
2. **Other Requirements**

Appendix A: Glossary Appendix S: Analysis Models

## Introduction

The aim of Pantry2Plate is to provide Recipes based on ingredients we have at home. My Pantry gives you the best recipes as a result that match your ingredients and dietary preferences.

Select ingredients from a dynamic list, or enter it in your own and watch recipes update instantly.

The recipes are easy to follow as some have

 Detailed Explanation of it and some even have

 Step-by-step videos solution of it.

This website P2P is the complete solution for all cooking-related problems.

These all features will be provided in a single platform which will be very beneficial and time-saving for all.

#### Purpose

* + - Recipe Inspiration
    - Reducing Food Waste
    - Convenience
    - Time-Saving

#### Definitions, Acronyms, and Abbreviations.

* **Recipe Book**: A collection of instructions for preparing dishes, typically organized by categories such as cuisine type, meal type, or ingredient.
* **Web Page**: A single page on the World Wide Web, often containing information or interactive elements.
* **UI: (User Interface):** The visual or interactive components of a webpage that allow users to interact with and navigate the content.
* **UX: (User Experience):** The overall experience a user has while interacting with a webpage, including ease of use, functionality, and satisfaction.
* **API: (Application Programming Interface):** A set of rules and protocols that allows one software application to interact with another. In this context, it might be used to communicate with a database of recipes or ingredients.
* **Search Bar**: An input field on a webpage where users can enter keywords or ingredients to search for specific recipes.
* **Filter**: A feature that allows users to narrow down recipe search results based on criteria such as cuisine, dietary restrictions, or cooking time.
* **Ingredient List**: A list of all the ingredients required to prepare a specific recipe.
* **Step-by-Step Instructions**: Detailed guidance on how to prepare a dish, typically broken down into individual steps.
* **Yield**: The number of servings a recipe will make.
* **Prep Time**: The estimated amount of time required to prepare ingredients before cooking.
* **Cook Time**: The estimated amount of time required to cook the dish.
* **Total Time**: The sum of prep time and cook time, representing the overall time required to make the dish.
* **Nutritional Information**: Details about the nutritional content of a recipe, including calories, fat, protein, and more.
* **Bookmark**: A feature that allows users to save their favorite recipes for future reference.
* **Favorites**: A section or feature where users can access their saved or favorite recipes.
* **Shopping List**: A feature that generates a list of ingredients based on selected recipes, making it easier for users to plan their grocery shopping.
* **Gluten-Free (GF)**: Indicates that a recipe does not contain gluten, a protein found in wheat, barley, and rye.
* **Vegetarian (V):** A recipe that does not include meat or fish.
* **Vegan** **(VG):** A recipe that excludes all animal products, including dairy and eggs.
* **Paleo**: A dietary approach that emphasizes whole foods and avoids processed foods, dairy, and grains.
* **Keto or Ketogenic**: A low-carbohydrate, high-fat diet that aims to induce a state of ketosis in the body.
* **GFAP (Gluten-Free All-Purpose) Flour**: A type of flour suitable for gluten-free baking and cooking.
* **EVOO** **(Extra Virgin Olive Oil):** A common abbreviation for high-quality olive oil.
* **Tbsp** or **tbsp**: Abbreviation for tablespoon, a unit of volume measurement in cooking.
* **Tsp** or **tsp**: Abbreviation for teaspoon, a smaller unit of volume measurement in cooking.
* **oz or oz.:** Abbreviation for ounce, a unit of weight measurement in cooking.
* **°C or °F**: Symbols for degrees Celsius and degrees Fahrenheit, used to indicate temperature settings for cooking.
* **Crockpot:** A brand name often used as a generic term for a slow cooker.
* **Mixer:** A kitchen appliance used to blend, beat, or mix ingredients.
  1. ***References***

The references for the above software are as follows : -

i. www.google.co.in

1. [www.winkipedia.com](http://www.winkipedia.com/)
2. IEEE. Software Requirements Specification Std. 830-1993.
3. Chevy Chase Bank, UMBC Branch.
4. Russell C. Bjork Requirements Statement for Example ATM System. Online.

URL: <http://www.math-cs.gordon.edu/local/courses/cs211/ATMExample/>

#### Overview

Section 1.0 discusses the purpose and scope of the software. Section 2.0 describes the overall functionalities and constraints of

the software and user characteristics.

Section 3.0 details all the requirements needed to design the software.

1. **The Overall Description**

#### Product Perspective

* **Integration with Existing Resources**: The recipe book or webpage should seamlessly integrate with existing culinary resources, such as cookbooks, cooking apps, and other recipe websites. It should complement these resources by offering a unique feature, such as ingredient-based recipe recommendations.
* **User-Centric Design**: The product should prioritize user experience and usability. It should feature an intuitive user interface (UI) that allows users to easily input their available ingredients, search for recipes, and access step-by-step instructions.
* **Database Integration**: The product likely relies on a database of recipes and ingredients. It should ensure that this database is regularly updated to include a wide variety of recipes and ingredients to cater to diverse user preferences.
* **Compatibility**: The product should be compatible with various devices and web browsers to reach a broad user base, including those using smartphones, tablets, and desktop computers.
* **Customization**: Users should have the ability to customize their experience, such as setting dietary preferences (e.g., vegetarian, vegan, gluten-free) and portion sizes to receive tailored recipe recommendations.
* **Filtering and Sorting:** The product should provide filtering and sorting options to help users refine search results based on criteria like cooking time, cuisine type, or popularity.
* **Ingredient Availability Check**: The system should ideally include a feature to check ingredient availability based on the user's location, which can be especially helpful for global users with access to different ingredients.
* **Recipe Rating and Reviews:** Users should have the ability to rate recipes and provide reviews. This enhances the credibility of the recipes and helps others make informed decisions.
* **Bookmarking and Saving**: The product should allow users to save their favorite recipes for future reference. This feature enhances user engagement and retention.
* **Monetization and Partnerships**: Consideration should be given to potential revenue streams, such as advertisements, premium features, or partnerships with food brands and suppliers.
* **Security and Privacy:** User data, especially personal information and dietary preferences, should be securely stored and protected in compliance with relevant data protection regulations.
* **Scalability**: The product should be designed to handle increasing user traffic and a growing database of recipes and ingredients.
* **Support and Updates**: Continuous support, bug fixes, and updates are essential to ensure the product remains functional and competitive in the long term.
* **Competitive Analysis**: The product should be aware of and responsive to changes in the competitive landscape, adjusting its features and user experience accordingly.
* **Feedback Mechanism**: An avenue for user feedback should be provided to gather suggestions and address any issues users may encounter.
  1. ***Product Functions***

The major functions that **Pantry2Plate** performs are described as follows:-

* **Ingredient Input**: Users can input the ingredients they have on hand. This can be done through manual entry or by scanning barcodes (if available).
* **Recipe Search**: The system searches its database for recipes that can be made using the specified ingredients.
* **Recipe Recommendations**: Based on the ingredients entered, the product provides a list of recommended recipes that match the user's available ingredients.
* **Filtering and Sorting:** Users can filter and sort recipe results based on criteria such as cuisine type, dietary preferences, cooking time, or difficulty level.
* **Recipe Details:** Users can click on a recipe to view detailed information, including ingredients, step-by-step instructions, prep time, cook time, serving size, and nutritional information.
* **Recipe Ratings and Reviews:** Users can see ratings and read reviews from other users who have tried the recipe.
* **Save and Bookmark:** Users have the option to save recipes they like to their personal recipe collections or bookmark them for later reference.
* **Shopping List Generation:** The product can generate a shopping list based on the ingredients required for selected recipes, helping users plan their grocery shopping.
* **Customization:** Users can customize their profile by specifying dietary preferences, allergies, or dietary restrictions to receive more relevant recipe recommendations.
* **User Accounts:** Users can create accounts to save their settings, favorite recipes, and personal information securely.
* **Social Sharing:** Users can share their favorite recipes or cooking experiences on social media platforms.
* **Cross-Platform Access:** The product is accessible on multiple platforms, including web browsers, mobile apps, and possibly smart kitchen devices.
* **Recipe Submission:** Allow users to submit their own recipes to contribute to the community and expand the recipe database.
* **User Feedback and Support:** Provide a way for users to contact customer support, report issues, and offer feedback to improve the product.
* **Monetization:** Implement revenue-generating features, such as sponsored recipes, premium subscriptions for an ad-free experience, or partnerships with food brands.
* **Localization:** Consider providing recipes and content in multiple languages and adapting to different culinary traditions and measurement systems.
* **Search History:** Keep a history of users' searches and recipes they've viewed for easy reference.
* **Suggested Ingredient Swaps:** Offer suggestions for ingredient substitutions in case users don't have a specific ingredient listed in a recipe.
* **Video Tutorials:** Optionally, provide video tutorials or cooking tips to assist users in preparing recipes.
* **Integration with Smart Kitchen Devices:** For advanced users, consider integration with smart kitchen appliances to automate cooking processes.

#### User Characteristics

There are different kind of users that will be interacting with the system. The intended user of the software are as follows:-

* + - **User A:**A novice ATM customer. This user has little or no experiencewith electronic means of account management and is not a frequent user of the product. User A will find the product easy to use due to simple explanatory screens for each ATM function. He is also assisted by an intarctive teaching mechanism at every atep of the transaction, both with the help of visual and audio help sessions.
    - **Maintenance Personnel:**A bank employee. This user is familiar with thefunctioning of the ATM. This user is in charge of storing cash into the ATM vault and repairing the ATM in case of malfunction. This user is presented with a different display when he logs in with the admninistrator’s password and is provided with options different from that of normal user. He has the authority to change or restrict various features provided by the software in situations of repairing.

#### Constraints

The major constraints that the project has are as follows:-

* + - The ATM must service at most one person at a time.
    - The number of invalid pin entries attempted must not exceed three. After three unsuccessful login attempts, the card is seized/blocked and need to be unlocked by the bank.
    - The simultaneous access to an account through both, the ATM and the bank is not supported.
    - The minimum amount of money a user can withdraw is Rs 100/- and the maximum amount of money a user can withdraw in a session is Rs.10,000/- and the maximum amount he can withdraw in a day is Rs 20,000/-
    - Before the transaction is carried out, a check is performed by the machine to ensure that a minimum amount of Rs 1000/- is left in the user’s account after the withdrawal failing which the withdrawal is denied.
    - The minimum amount a user can deposit is Rs 100/- and the maximum amount he can deposit is Rs 10,000/-.
    - A user can select only that cellular operator for mobile bill clearings that is supported by the bank.
    - The software requires a minimum memory of 20GB
    - The databse used should be Oracle7.0.
    - There shall be a printer installed with the machine to provide the user with the printed statement of the transaction.
    - For voice interactions, speakers should also be there to accompany the machine.

#### Assumptions and Dependencies

The requirements stated in the SRS could be affected by the following factors:

* + - One major dependency that the project might face is the changes that need to be incorporated with the changes in the bank policies regarding different services. As the policies changes the system needs to be updated with the same immediately. A delay in doing the same will result to tremendous loss to the bank. So this should be changed as and when required by the developer.
    - Another constraint relating to the operating environment is that we are specific to Oracle Database.
    - The project could be largely affected if some amount is withdrawn from the user’s account from the bank at the same time when someone is accessing that account through the ATM machine. Such a condition shall be taken care of.
    - At this stage no quantitive measures are imposed on the software in terms of speed and memory although it is implied that all functions will be optimized with respect to speed and memory.

It is furthermore assumed that the scope of the package will increase considerably in the future.

1. ***External Interface Requirement***

### User Interface Requirements

The interface provided to the user should be a very user-friendly one and it should provide an optional interactive help for each of the service listed. The interface provided is a menu driven one and the following screens will be provided:-

* + - 1. A login screen is provided in the beginning for entering the required username/pin no. and account number.
      2. An unsuccessful login leads to a reattempt(maximum three) screen for again entering the same information. The successful login leads to a screen displaying a list of supported languagesfrom which a user can select any one.
      3. In case of administrator, a screen will be shown having options to reboot system, shut down system, block system, disable any service.
      4. In case of reboot/ shut down, a screen is displayed to confirm the user’s will to reboot and also allow the user to take any backup if needed.
      5. In case of blocking system, a screen is provided asking for the card no. By entering the card no of a particular user, system accedes can be blocked for him.
      6. Administrator is also provided with a screen that enables him to block any service provided to the user by entering the name of the service or by selecting it from the list displayed.
      7. After the login, a screen with a number of options is then shown to the user. It contains all the options along with their brief description to enable the user to understand their functioning and select the proper option.
      8. A screen will be provided for user to check his account balance.
      9. A screen will be provided that displays the location of all other ATMs of same bank elsewhere in the city.
      10. A screen will be provided for the user to perform various transactions in his account.

The following reports will be generated after each session dealt with in the machine:-

1. The login time and logout time along with the user’s pin no and account number is registered in the bank’s database.
2. The ATM’s branch ID through which the session is established is also noted down in the bank’s database.
3. Various changes in the user’s account after the transactions,if any, are reported in the database.
4. A printed statement is generated for the user displaying all the transactions he performed.

Other various user interface requirements that need to be fulfilled are as follows:-

* + The display screen shall be of 10" VGA color type.
  + The display screen shall have 256 color resolution.
  + The display screen shall also support touchscreen facility.
  + The speakers shall support Yamaha codecs.
  + The keypad shall consist of 16 tactile keys.
  + There shall be 8 tactile function keys.
  + The keyboard will be weather resistant.
  + The transaction receipt shall be 3.1" × 6".
  + The statement receipt shall be 4.2" × 12".
  + The deposit envelopes shall be 9" long and 4" wide.
    1. **Hardware Interface Requirements**

There are various hardware components with which the machine is required to interact. Various hardware interface requirements that need to be fulfilled for successful functioning of the software are as follows:-

* The ATM power supply shall have a 10/220 V AC manual switch.
* The ATM card should have the following physical dimensions:-

|  |  |  |  |
| --- | --- | --- | --- |
| o | Width | - | 85.47mm-85.72mm |
| o | Height | - | 53.92mm-54.03mm |
| o | Thickness | - | 0.76mm+0.08mm |

* The card reader shall be a magnetic stripe reader
* The card reader shall have Smart card option.
* The slot for a card in thye card reader may include an extra indentation for the embossed area of the card. In effect it acts as a polarization key and may be used to aid the correct insertion orientation of the card. This is an additional characteristic to the magnetic field sensor which operates off the magnetic stripe and is used to open a mechanical gate on devices such as ATMs.
* There shall be a 40 column dot matrix receipt printer.
* There shall be a 40 column dot matrix statement printer.
* The receipt dispenser shall be a maximum of 4" width and 0.5" thickness.
* The statement dispenser shall be a maximum of 5" width and 0.5" thickness.
* The envelope depository shall be a maximum of 4.5" width, 10" length and 0.5" thickness.
* Screen resolution of at least 800X600-required for proper and complete viewing of screens. Higher resolution would not be a problem.
  + 1. **Software Interface Requirements**

In order to perform various different functions, this software needs to interact with various other softwares. So there are certain software interface requirements that need to be fulfilled which are listed as follows:-

* The transaction management software used to manage the transaction and keep track of resources shall be BMS version 2.0.
* The card management software used to verify pin no and login shall be CMS version 3.0.
* Yamaha codecs 367/98 for active speakers.
* The database used to keep record of user accounts shall be Oracle version7.0.
  + 1. **Communication Interface Requirements**

The machine needs to communicate with the main branch for each session for various functions such as login verification, account access etc. so the following are the various communication interface requirements that are needed to be fulfilled in order to run the software successfully:-

* The system will employ dial-up POS with the central server for low cost communication.
* The communication protocol used shall be TCP/IP.
* Protocol used for data transfer shall be File Transfer Protocol.(FTP)

1. ***System Features***

### Remote Banking and Account Management

**Description**

The system is designed to provide the user with the facility of remote banking and perform various other functions at an interface without any aid of human bank teller. The functioning of the system shall be as follows:-

At the start, the user is provided with a log in screen and he is required to enter his PIN NO. and Account details which are then verified by the machine. In case of an unsuccessful attempt a user is asked again for his credentials but the maximum number of attempt given to the user is limited to 3 only, failing which his card is blocked and need to be unblocked by the bank for any future use.

After a successful log in, the user is presented with a list of language. The user can select any one in the list for interaction with the machine for the

entire session. After the language selection the user is also asked whether he wants to fix that language for future use also so that he is never asked for language in future. In addition there is also a facility for the user to switch to any other language during that session.

After the language selection, the user is directed towards a main page that displays a set of options/services along with their brief description, enabling the user to understand their functioning. The user can select any of the listed option and can continue with the transaction.

The machine also provides the user with a number of miscellaneous services such as:

The machine lists a set of operators that are supported by the bank. A user can clear off his pending mobile phone bills be selecting his operator.

The machine also has the facility to display a map that marks the location of other ATMs of the same bank in the city. This may help the user to look for the ATM nearest to his destination.

At any moment if the user wants to abort the transaction, he is provided with an option to cancel it. Just by pressing the abort button he can cancel all the changes made so far and can begin with a new transaction.

After the user is finished with his work, for security purpose, he is required to log out and then take his card out of the slot.

### Validity Checks

In order to gain access to the system, the user is required to enter his/her correct user id/pin no and account no failing which his card may be blocked.

The user can access only one account at a time and can enter only one account no.

Also if the user is an administrator, he is required to enter his login id in order to access and change the facilities provided by the system.

### Sequencing Information

The information about the users and their account should be entered into the database prior to any of the transactions and the backup be maintained for all account information.

### Error Handling/ Response to Abnormal Situations

If any of the above validation/sequencing flow does not hold true, appropriate error messages will be prompted to the user for doing the needful.

#### Receipt Generation

After ech transaction user has performed, a receipt is generated that contains all the information about the transaction. The format of the generated receipt is as shown below:-

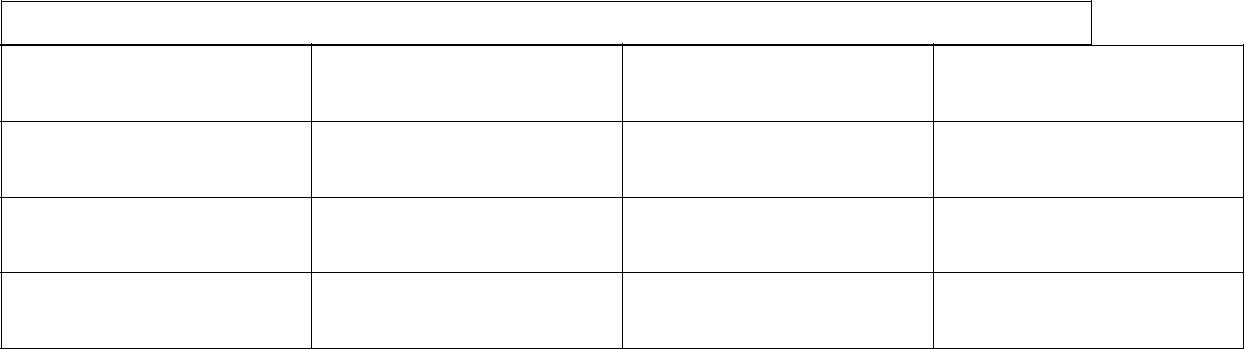
### KPM BANK

**Branch name/Id (address)**

### Login Time:- Date:- Account No:-

**User Name:-**

### TRANSACTIONS:



**FROM TO TYPE AMOUNT**

**Logout Time:- BARCODE**

### Thank You For your visit.

**See you soon.**

1. ***Other Non-functional Requirements***

***5.1 Performance Requirements***

The following list provides a brief summary of the performance requirements for the software:

### Capacity

* + - * The Pantry2Plate shall provide customers 24x7 hour service.

### Dynamic requirements

* + - * The card verification time must not exceed 0.8 sec. under normal server workload and 1 sec. under peak server workload.
      * The pin number verification time must not exceed 0.3 sec. under normal server workload and 0.5 sec. under peak server workload.
      * Account balance display time must not exceed 2 sec. under normal server workload and 3 sec. under peak server workload.
      * Account balance transfer time must not exceed 3 sec. under normal server workload and 4 sec. under peak server workload.
      * Cash withdrawal transaction time must not exceed 4 sec. under normal server workload and 5 sec. under peak server workload.
      * Deposit transaction time after insertion of the deposit envelope must not exceed 5 sec. under normal server workload and 6 sec. under peak server workload.
      * Receipt printing time after must not exceed 3 sec. under normal server and peak server workload.
      * Touch screen and button response time must not exceed 5000ms.
      * Credit card advance time must not exceed 6 sec. under normal traffic and server and peak traffic and server workload.
    1. **Quality** –The primary objective is to produce quality software. Asthe quality of a piece of software is difficult to measure quantitatively, the following guidelines will be used when judging the quality of the software:

1. Consistency – All code will be consistent with respect to the style. (This is implied when adhering to the standard).
2. Test cases – All functionality will be thoroughly tested

#### Software System Attributes

* + 1. **Reliability**
* The algorithm used to match ingredients with recipes should be reliable and produce accurate results. Test the algorithm thoroughly to avoid incorrect recommendations
  + 1. **Availability**
* Maintain high system availability to ensure users can access recipes whenever they need them. Downtime can frustrate users and disrupt meal planning.
* Employ redundancy and failover strategies to minimize downtime due to server or network issues

### Security

* Prioritize security measures to protect user data, especially personal information and dietary preferences.
* Implement authentication and authorization mechanisms to ensure that only authorized users can access specific features or data.
* The system shall be compatible with AIMS security standards.
* The Encryption standard used during pin transmission shall be triple DES.
* The password shall be 6-14 characters long.
* Passwords shall not contain name of customers as they are easy to be hacked.
* Passwords can contain digit, hyphen and underscore.
* User should be provided with only three attempts for login failing which his account needs to be blocked.
  + 1. **Maintainability**
* The system must accurately recognize and match user-input ingredients with ingredients in the recipe database.
* Ingredients with different names (e.g., "bell peppers" and "capsicum") should be matched correctly.

#### Business Rules

The business rules for the software are as follows:

* **User Account Creation:**

Users can create accounts with a valid email address and password.

Usernames and passwords must meet specified security requirements (e.g., minimum length, complexity).

* **User Profile:**

Upon registration, users can complete their profiles by adding optional information such as a profile picture, dietary preferences, and a brief bio.

* **Recipe Submission:**

Registered users can submit their own recipes, including ingredients, instructions, and optional images.

Submitted recipes should be reviewed by administrators for accuracy and appropriateness before being published.

* **Recipe Editing:**

Users can edit or update their submitted recipes as needed.

Changes made to a recipe should be tracked to maintain transparency.

* **Recipe Ownership:**

Users retain ownership of the recipes they submit, but the platform may use submitted recipes for promotional purposes with proper attribution.

* **Recipe Privacy Settings:**

Users can choose to make their recipes public (visible to all users) or private (only visible to themselves).

Users can also set recipes as shareable with specific users or groups.

* **User Recipe Feed:**

Registered users have a personalized recipe feed that displays recipes from other users they follow or interact with.

Algorithms may be used to suggest recipes based on user preferences and activity.

* **Recipe Comments and Ratings:**

Allow users to leave comments and ratings on recipes to provide feedback and encourage community interaction.

* **Recipe Favourites and Collections:**

Users can save recipes to their "favourites" and create custom collections or folders to organize recipes for easy access.

* **Recipe Sharing:**

Users can share recipes on social media platforms or via email to promote their recipes to a wider audience.

* **Monetization Opportunities:**

Explore monetization options, such as premium memberships, ad revenue sharing with content creators, or sponsored content.

* **Content Ownership and Copyright:**

Ensure that user-submitted recipes do not infringe on copyright or intellectual property rights. Encourage users to submit their original creations.

* **User Privacy and Data Security:**

Safeguard user data, including personal information, recipes, and activity history, in compliance with data protection regulations.

* **Terms of Service and Community Guidelines:**

Develop clear terms of service and community guidelines outlining acceptable behaviour, content standards, and consequences for violations.

* **Monetization Opportunities:**

Explore monetization options, such as premium memberships, ad revenue sharing with content creators, or sponsored content.

* **Content Ownership and Copyright:**

Ensure that user-submitted recipes do not infringe on copyright or intellectual property rights. Encourage users to submit their original creations.

* **User Privacy and Data Security:**

Safeguard user data, including personal information, recipes, and activity history, in compliance with data protection regulations.

* **Terms of Service and Community Guidelines:**

Develop clear terms of service and community guidelines outlining acceptable behaviour, content standards, and consequences for violations.

### Other Requirements

**Appendix A: Glossary**

* **Ingredient Matcher**: The algorithm or system component responsible for identifying and suggesting recipes based on user-input ingredients.
* **Recipe Database**: The collection of recipes stored within the platform, including ingredients, cooking instructions, and user-contributed content.
* **User Profile**: A user's personal space on the platform, displaying information such as their username, profile picture, and dietary preferences.
* **Public Recipe**: A recipe that is visible to all users of the platform, accessible for viewing and cooking.
* **Private Recipe**: A recipe that is only visible and accessible to the user who uploaded it, ensuring privacy.
* **User Authentication**: The process of verifying user identities securely to prevent unauthorized access to accounts and personal information.
* **Shopping List Generator**: A feature that generates a shopping list based on the ingredients required for selected recipes, aiding users in planning their grocery shopping.
* **Data Backup and Recovery**: Processes and mechanisms for regular data backups and the restoration of data in case of data loss or system failures.
* **Data Privacy Measures**: Safeguards and practices implemented to protect user data, including personal information and user activity history, in compliance with data protection regulations.
* **Terms of Service**: A legal document outlining the rules and conditions governing the use of the platform by its users.
* **Recipe Ownership**: The concept that users retain ownership of the recipes they submit while granting the platform permission to use them for promotional purposes with proper attribution.
* **Recipe Categories**: Classifications used to organize recipes based on attributes like cuisine type, meal category (e.g., breakfast, dinner), and dietary restrictions.
* **Recipe Privacy Settings**: User-configurable options that determine the visibility of their uploaded recipes (public, private, or shareable with specific users or groups).

|  |  |  |
| --- | --- | --- |
|  |  | around the world via the TCP/IP protocol. |
| **Smart Card** | **-** | Card without hardware which stores the user’s private keys |
|  |  | within a tamper proof software guard. |
| **Tactile -** | Special keyboard designed to aid the visually impaired. | |
| **Keyboard** |  |  |
| **TCP/IP** | **-** | Transmission Control Protocol/Internet Protocol. |