

Online Toys Finding Application

**Software Requirements Specification**

Version 1.0



**Group Id: F230262DF5(BC200406111)**

**Supervisor Name: Akmal khan**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| **20/11/2023** | 1.0 | **SRS stand for “software requirement specification”. This is description of overall Applications development. This is a starting point of Application development OR project. In “SRS” document we prepared scope of project. We also create the use case diagram demonstrate the customer of application. We are creating the project plan and discuss the development methodologies in SRS document. SRS leads the abstract idea to its concretion.** | **BC200406111** |

**Table of Contents**

1. [Scope (of the project)](#scope)
2. [Functional Requirements Non Functional requirements](#FRNFR)
3. [Use Case Diagram](#UCD)
4. [Usage Scenarios](#UCS)
5. [Adopted Methodology](#Adopted)
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

|  |
| --- |
| **SRS Document:** |

**Scope of Project:**

In “online toys finding web application” the user will be able to search for the particular toys and purchase it, Then the “Admin” will deliver the toy(s) to the user and user take it safely. Online toys finding web application will make it easy for customers to buy the toys and order it. And other home related items by ordering from their home. As a we know that today is the age of e-commerce, and we know that many kinds of web application is a part of life. Online toys finding web application will be another addition for local e-commerce and make it as a possible to buy and easily order it.

|  |
| --- |
| **Functional and non-Functional Requirements** |

* **Functional** **Requirements:**

**Admin:**

1. Admin can login/logout.
2. Admin can add, update and delete different categories of toys.
3. Admin can add, update and delete different toys information such as toy name, picture, price, quality etc.
4. Admin can view the order of the user in which all details should be visible to the admin like, selected Toys, total price, shipping address etc.
5. Admin can generate voucher for the user.
6. Admin can verify the user online dues or paid vouchers and after the verification, the Admin can deliver the order.
7. Admin can add, update, and delete the shipping charges city wise and when the user provides city name in the shipping address, the shipping charges should automatically have added with the total price of the order.
8. Admin can view the user request for a particular toy and can provide feedback accordingly.
9. Admin can manage the stock.
10. Admin can manage expense reports.
11. Admin can manage profit reports.

**User/Parent:**

1. User can register.
2. User can login/logout.
3. User can update his/her profile.
4. User can search the toys as per his/her choice.
5. User can view all the available toys information such as toy picture, its category, price, etc.
6. User can send a request for a particular toy (if not available). At the time of request, the user should provide some information like toy name, picture (if available) and further description of the toy.
7. User can add as many as per his/her choice toys to the shopping cart.
8. User can add or delete any item from the shopping cart.
9. The Total Price of toys should be automatically adjusted after the addition or deletion of toys in the shopping cart
10. User should provide the complete shipping address, when the user confirms the order.
11. User can pay the total dues online or the user can download the voucher and paid it and then upload the paid voucher.
12. User can check the status of his/her order.

* **Non Functional Requirements:**

1. **Security**

* User will access his/her profile after login through the user id and password
* Details of user must be secure and safe

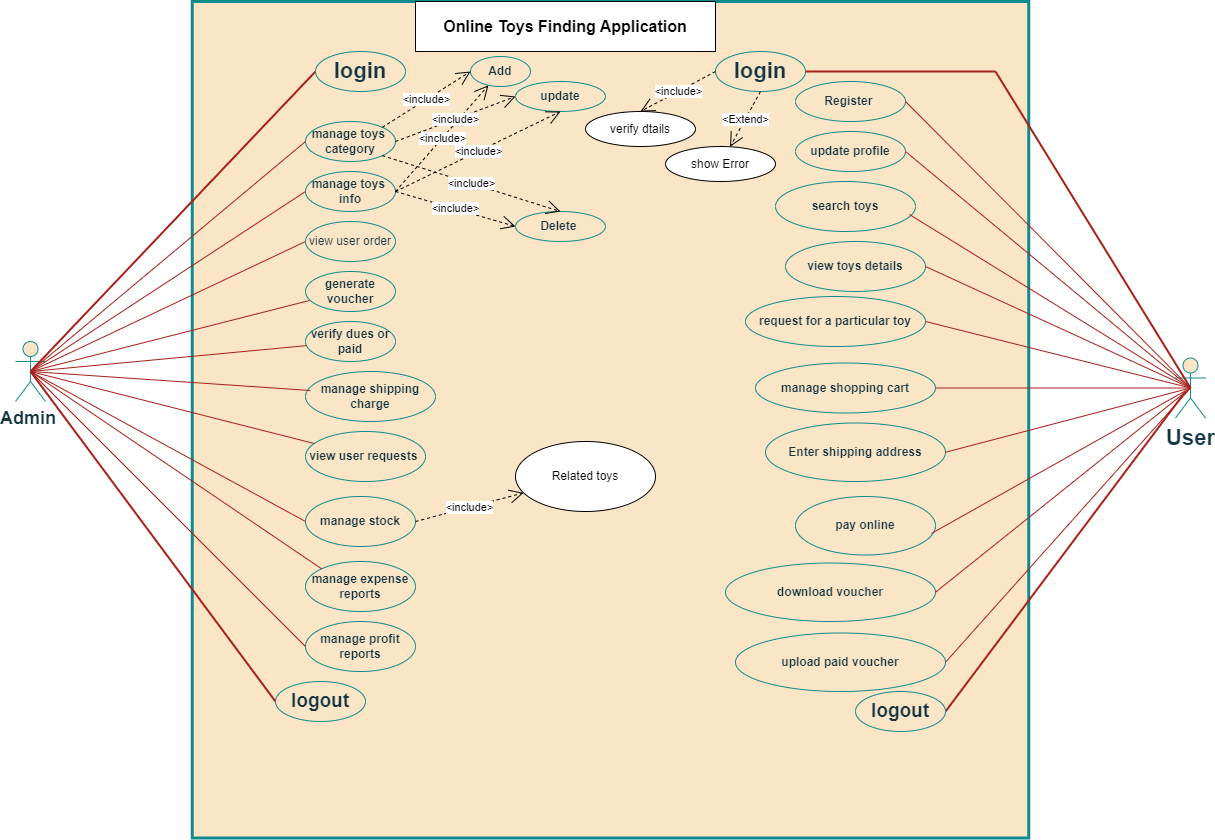
1. **Performance**

* Web application should be lightweight, so that it can run smoothly even on the slow internet

1. **Reusability**
   * Database must be backup those who are purchase show it.
   * User must be valid.
2. **Portability**

* It run on different platform, and different internet speed.

|  |
| --- |
| **Use case diagram** |



|  |
| --- |
| **Usage scenarios for user/parent:** |

|  |  |
| --- | --- |
| **Use case title** | Login |
| **Use case id** | UC-id-01 |
| **Action** | User enters Email, password and click “login” |
| **Description** | Allows a user to access the web application by providing valid login Email and password. |
| **Alternative Paths** | 1. User forget password 2. User account is invalid. |
| **Pre -Conditions** | User has a valid account enter. |
| **Post Conditions** | User is login in web application display the user home page. |
| **Exceptions** | 1. Invalid 2. Internet error |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Register |
| **Use case id** | UC-id-02 |
| **Action** | User fills out registration form and clicks "Register" |
| **Description** | Allows a user to create a new account in the application. |
| **Alternative Paths** | 1. User enters existing email 2. Enter the Password |
| **Pre -Conditions** | User is not registered in the application. |
| **Post Conditions** | 1. User is registered 2. And application redirects to login page |
| **Exceptions** | 1. Existing email 2. password 3. Internet error |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Update profile. |
| **Use case id** | UC-id-03 |
| **Action** | User setting the profile and modifies it. |
| **Description** | Allows a user to update their profile information such as name, email, or password |
| **Alternative Paths** | 1. 1. User wants to change the profile picture 2. User updates contact information |
| **Pre -Conditions** | User is logged in |
| **Post Conditions** | User profile information is updated |
| **Exceptions** | 1. Invalid data format 2. Password change fails 3. Internet error |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Search toys |
| **Use case id** | UC-id-04 |
| **Action** | User enters toy name and clicks "Search". |
| **Description** | Allows a user to find toys all kinds of toys. |
| **Alternative Paths** | 1. 1. Categories of toys. 2. User applies multiple toys. |
| **Pre -Conditions** | User is on the home page of toys. |
| **Post Conditions** | Display a list of toys matching the search criteria |
| **Exceptions** | 1. No toys found 2. Internet error |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | View toys details |
| **Use case id** | UC-id-05 |
| **Action** | User clicks on a specific toy to view details |
| **Description** | Allows a user to see detailed information about a selected toy |
| **Alternative Paths** | 1. User searches for a toy 2. User navigate from homepage to a specific toy |
| **Pre -Conditions** | User is logged in and on the homepage |
| **Post Conditions** | User see detailed information about the select toy |
| **Exceptions** | 1. Selected toy not found 2. Network error |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Logout |
| **Use case id** | UC-id-06 |
| **Action** | User clicks on "Logout" button |
| **Description** | Allows a user to log out of the web application |
| **Alternative Paths** | 1. User is inactive for a specific period 2. User manually clicks on the browser's "Logout" option |
| **Pre -Conditions** | User is logged in |
| **Post Conditions** | User is logout, and the application redirects to the home page for login |
| **Exceptions** | 1. Session timeout 2. Network issues 3. Internet error |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Request for particular toy |
| **Use case id** | UC-id-07 |
| **Action** | User searches for a specific toy using search bar. |
| **Description** | At the time of request, the user should provide some information like toy name, picture (if available) and further description of the toy. |
| **Alternative Paths** | 1. User search for particular toys 2. User clicks on a suggested toy category |
| **Pre -Conditions** | User is on the home/search page |
| **Post Conditions** | Application displays search results with matching toys |
| **Exceptions** | 1. No matching results found 2. Internet error during search |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Manage shopping cart |
| **Use case id** | UC-id-08 |
| **Action** | User adds, removes, or updates items in the shopping cart |
| **Description** | User can add, remove and update as many as per his/her choice toys to the shopping cart. |
| **Alternative Paths** | 1. User updates quantity of items 2. User removes an items. |
| **Pre -Conditions** | User is login, and there are items in the shopping cart |
| **Post Conditions** | Shopping cart is updated according to user actions |
| **Exceptions** | 1. Item is out of stock 2. Internet error during update 3. User attempts to remove non-existent item |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Enter shipping address |
| **Use case id** | UC-id-09 |
| **Action** | User provide the complete shipping address, when the user confirm the order. |
| **Description** | Allows the user to input the shipping address for the selected toys |
| **Alternative Paths** | 1. User edits the shipping address before confirming 2. User selects a saved address from their profile |
| **Pre -Conditions** | User has items in the shopping cart |
| **Post Conditions** | Shipping address is stored for the order |
| **Exceptions** | 1. Incomplete address 2. Invalid characters in the address 3. Network connection issues |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Pay online |
| **Use case id** | UC-id-10 |
| **Action** | User selects toys for purchase and clicks "Proceed to Checkout" "Pay Online" or not |
| **Description** | Allows a user to make an online payment for selected toys |
| **Alternative Paths** | 1. User decides to continue shopping and adds more toys 2. User cancels the order before payment |
| **Pre -Conditions** | User has items in the shopping cart |
| **Post Conditions** | Payment is successfully processed, and the order is confirmed |
| **Exceptions** | 1. Insufficient funds 2. Payment gateway error 3. User cancels payment |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Download voucher |
| **Use case id** | UC-id-11 |
| **Action** | User requests to download a voucher. |
| **Description** | This use case represents the scenario where a user requests to download a voucher from the application |
| **Alternative Paths** | 1. User cancels the download during the process. 2. And again download |
| **Pre -Conditions** | User is authenticated and authorized to download vouchers. |
| **Post Conditions** | Voucher is successfully downloaded and saved by the user. |
| **Exceptions** | 1. Internet error during the download process. 2. Time out |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Upload paid voucher |
| **Use case id** | UC-id-12 |
| **Action** | User uploads a paid voucher. |
| **Description** | This use case involves the user upload a voucher on the application after payment. |
| **Alternative Paths** | 1. Voucher format is invalid. 2. If the payment is not confirmed, display an error message. |
| **Pre -Conditions** | User is login. Payment for the voucher is successful. |
| **Post Conditions** | Voucher is successfully uploaded and linked to the user account. |
| **Exceptions** | 1. If the voucher file is corrupted, display an error message. 2. If the user is not login, redirect to the login page. 3. If payment fails, display an error message. |
| **Author** | Bc200406111 |

|  |
| --- |
| **Usage Scenarios for Admin:** |

|  |  |
| --- | --- |
| **Use case title** | Manage toys category. |
| **Use case id** | UC-id-01 |
| **Action** | Admin selects "Manage Toys Category" option from the admin dashboard |
| **Description** | This use case allows the admin to perform actions related to the management of toy categories, such as add, update and delete categories of toys. |
| **Alternative Paths** | Admin select the for add, delete and update the toys. |
| **Pre -Conditions** | Admin is login and has the necessary permissions. Toys categories exist in the application. |
| **Post Conditions** | Updated toys categories are saved in the application. |
| **Exceptions** | 1. Invalid input for category details. 2. Application error during category update. |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | View user order. |
| **Use case id** | UC-id-02 |
| **Action** | Admin see the user orders. |
| **Description** | An admin selects a user and views their order history. |
| **Alternative Paths** | Admin selects a different user and see details. |
| **Pre -Conditions** | Admin is login and has appropriate permissions. |
| **Post Conditions** | Admin successfully views the user's order history. |
| **Exceptions** | Error message displayed if user has no orders. |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Generate voucher. |
| **Use case id** | UC-id-03 |
| **Action** | Admin generates a voucher for a user or a group of users. |
| **Description** | The admin initiates the process to create a voucher with specified details such as amount, validity, and any restrictions. |
| **Alternative Paths** | N/A |
| **Pre -Conditions** | Admin must be login to the application |
| **Post Conditions** | A voucher is generated and stored in the application. |
| **Exceptions** | 1. Admin not login 2. Admin not generate the voucher for user. |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Verify due and paid. |
| **Use case id** | UC-id-04 |
| **Action** | Admin verifies payments dues or paid. |
| **Description** | Admin checks the application for due payments and verifies if they have been paid then ok and give the toys. |
| **Alternative Paths** | N/A |
| **Pre -Conditions** | Admin is login |
| **Post Conditions** | Admin has verified due and paid successfully |
| **Exceptions** | If application is unable to retrieve payment status. |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Manage shipping charge. |
| **Use case id** | UC-id-05 |
| **Action** | Admin sets or updates shipping charges for products. |
| **Description** | Admin can add, update, and delete the shipping charges city wise and when the user provide city name in the shipping address, the shipping charges should automatically added with the total price of the order. |
| **Alternative Paths** | Admin can set default charges for all orders if specific charges are not defined for a particular zone or order type. |
| **Pre -Conditions** | Admin is login |
| **Post Conditions** | Shipping charges are updated or deleted as per admin's action. |
| **Exceptions** | 1. If the application encounters an error while updating or deleting shipping charges, an error message is displayed. 2. Admin may need to retry the operation. |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Logout |
| **Use case id** | UC-id-06 |
| **Action** | Admin clicks on "Logout" button |
| **Description** | Allows a Admin to logout of the web application |
| **Alternative Paths** | 1. Admin is inactive for a specific period 2. Admin manually clicks on the application "Logout" option |
| **Pre -Conditions** | Admin is logged in |
| **Post Conditions** | Admin is logout, and the application redirects to the home page for login |
| **Exceptions** | 1. Session timeout 2. Network issues 3. Internet error |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | View user request. |
| **Use case id** | UC-id-07 |
| **Action** | Admin selects "View User Request" option. |
| **Description** | Admin can view the user request for a particular toy and can provide feedback accordingly. |
| **Alternative Paths** | 1. Admin enters invalid. 2. Application error in fetching data |
| **Pre -Conditions** | User request exists |
| **Post Conditions** | Admin views user request details. |
| **Exceptions** | Invalid details. |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Manage stock. |
| **Use case id** | UC-id-08 |
| **Action** | Admin manages stock levels |
| **Description** | This use case allows the admin to manage the stock levels of toys in application. The admin can add new stock, update existing stock levels, and remove stock when necessary. |
| **Alternative Paths** | 1. Admin cancels stock update. 2. Admin decides not to add new stock. |
| **Pre -Conditions** | The admin is login to the Application. The toys inventory is accessible and up-to-date. |
| **Post Conditions** | The stock levels are updated according to the admin actions. The inventory records are synchronization. |
| **Exceptions** | 1. If the admin encounters a Application error while updating stock levels, an error message is displayed, and the changes are rolled back. 2. If the admin tries to update stock for a non-existing toys, 3. An error message is shown. |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Manage expense reports. |
| **Use case id** | UC-id-09 |
| **Action** | Admin login in the web application. |
| **Description** | The admin can access the expense management application to manage and review submitted expense reports. |
| **Alternative Paths** | N/A |
| **Pre -Conditions** | Admin login in application. |
| **Post Conditions** | Admin see the expenses reports. |
| **Exceptions** | If the admin cannot login. |
| **Author** | Bc200406111 |

|  |  |
| --- | --- |
| **Use case title** | Manage profit reports |
| **Use case id** | UC-id-10 |
| **Action** | Admin login in application. |
| **Description** | Admin can login in application and manage the report of profit related to toys profit. |
| **Alternative Paths** | N/A |
| **Pre -Conditions** | Admin is authenticated and has appropriate permissions |
| **Post Conditions** | Profit reports are successfully managed. |
| **Exceptions** | 1. Invalid login, 2. Insufficient permissions. |
| **Author** | Bc200406111 |

|  |
| --- |
| **Programming Methodologies** |

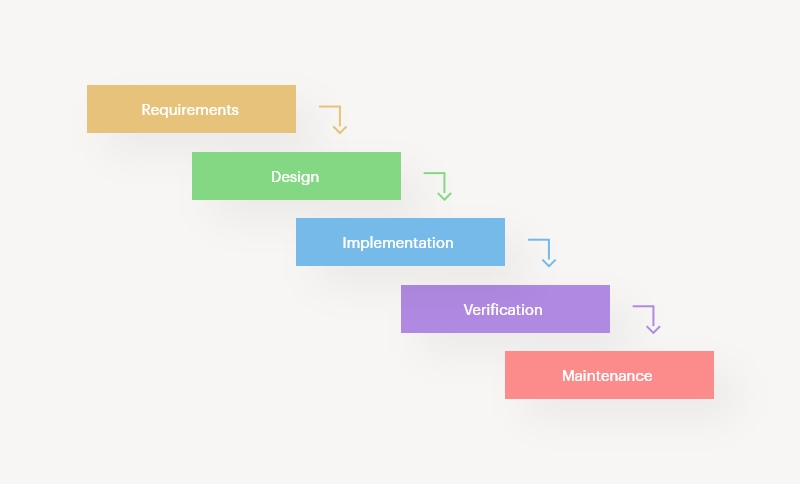
There are many different kinds of software or application development process model present in literature through which we can adopt one or combination of process model according of the need of our project. Some of the names of process models are mentioned below:

* Waterfall Model (This is the simplest and linear process model)
* Rapid Prototyping Model
* Spiral Model
* Fountain Model

**Waterfall Model** (The Traditional model)

Waterfall model is linear process model hence it is also called “Linear sequential model.” It suggests a sequence, systematic approach to software development that begins with system level and progress through the analysis, design, coding, testing and maintenance. In contrast to iterative models, each phase is runs through only once.

Following diagram depicts waterfall model:



* Reason of waterfall model:
* **Simple and Easy**

Waterfall model is simple and easy to understand.

* **Well-organized approach**

Many organizations like well-organized and disciplined approach hence they enjoy and prefer this well-structured approach.

* **Intuitive progress**

An advantage of this methodology is it so intuitive and allow you to start right away hence your project starts progress in linear way to the very end.

* **Clear goal**

If the project has clear goal, then waterfall model works beautifully. Waterfall model means that all the parts are joined together as one and go along with sole object of reaching of final result.

* **Heavy documentation**

Another great advantage of waterfall model is that it is heavy on documentation. When you have accurate and clear documentation then no any hurdle will be placed in your progress.

**Spiral Model:**

The spiral model is waterfall model plus risk analysis. In this model, every stage is proceeds by identification of alternatives and risk analysis and then further proceeds by evaluating and planning for the next phase. In this diagrammatic representation, it’s look like spiral with many loops



Reason of Spiral model:

* **Capability to handle risk**

High amount of risk analysis hence the rate of risk is reduced.

* **Great communication**

Customers is involved at every phase of development of project

* **Flexible in nature**

We can bring or make changes easily because it is flexible in nature.

* **Early production**

Software produced early in the life cycle.

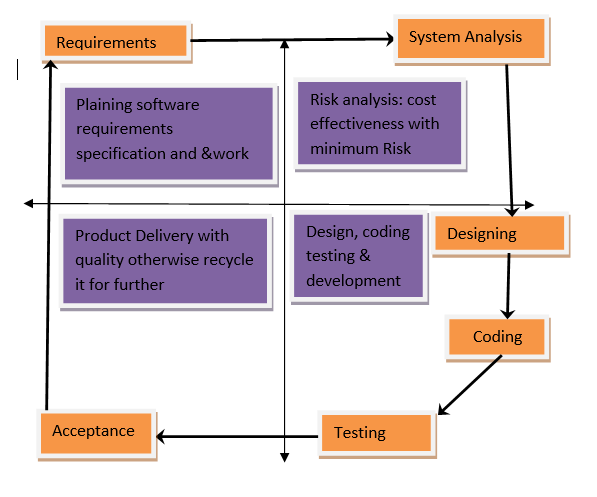
* **Good for complex projects**

Another great advantage of spiral model. It’s very good for large and complex projects.

|  |
| --- |
| **Adopt methodology** |

As per requirements we will adopt the methodology named. “VU process model”.

Here show vu process model.



The description which is given below.

**VU Process Model:**

* Vu process model is the combination of waterfall methodology and spiral methodology. It is also called hybrid approach of system development.
* In the requirement phase the Application goals, services and constrains are established after discussion with user OR The user give you requirements.
* The analysis phase is synchronized effort to find all the potential roadblocks and draft a robust plan to tackle each.
* This phase usually involves a lot of brainstorming within Application development team as well as with respective client.
* It maximizes the quality of Application and minimizes any risks and errors.

**Reasons for choosing this Methodology:**

Following are the basic reasons for choosing VU Process Model for our project.

* Our project is divided into different phases which we have to complete in sequence and submit to our supervisor.
* If some mistakes are found than he suggests us to correct that mistakes and improve our project. This process will be adopted due to its spiral nature.
* When the phase is well worked-out, and also accepted by our Supervisor then we will proceed to next phase. This will be done due to the waterfall nature of the VU Process model.
* Hence both models are suppressed in single form and that will be our VU process model. It will provide error free result as each step is done in a sequence.

**Work Plan/schedule**

