Business Requirements

Document

ONLINE JEWELRY

RENTAL SYSTEM

AUTHORS:

BARRIENTOS, Erica T.

ericabarrientoswork@gmail.com

RIZADA, Abdul Hanief N.

abdul.hanief.rizada@gmail.com

Table of Contents

[1](#_heading=h.gjdgxs) Executive Summary 3

[2](#_heading=h.30j0zll) Project Description 3

[3](#_heading=h.1fob9te) Project Scope 4

[3.1](#_heading=h.3znysh7) In Scope 4

[3.2](#_heading=h.2et92p0) Out of Scope 4

[4](#_heading=h.tyjcwt) Current Process 5

[5](#_heading=h.3dy6vkm) Functional Requirements 5

[5.1](#_heading=h.1t3h5sf) Priority 5

[6](#_heading=h.4d34og8) Glossary 6

[7](#_heading=h.2s8eyo1) References 6

[8](#_heading=h.17dp8vu) Appendix 8

[9](#_heading=h.3rdcrjn) Document History 9

# Executive Summary

This Business Requirements Document (BRD) outlines the requirements for the capstone project; Online Jewelry Rental System. This document is intended to display the requirements and solutions that must be done to fulfill the goal of the proposed system. The project is designed to lessen the time consumed by the customers that would benefit both parties; the company and the customers.

With this application, a customer can spend less time visiting a specific branch to look at or search for the jewelry they want to rent, and also for the administrator and employees to manage their product inventory more efficiently.

# Project Description

Nowadays, jewelry is basically one of the important parts of our attire, but purchasing new jewelry for every occasion can be pricey. Renting is therefore a wiser decision. It can take lots of time for someone to visit a store location to look for a nice-looking item that can be rented or bought. To address these issues, the developers decided to introduce an Online Jewelry Rental System to serve as a platform for someone who wanted to just search for a specific jewelry, or appoint a schedule to negotiate with the employees, or to rent without visiting the actual store.

This document contains both functional and non-functional requirements, an overview of the current process, as well as the proposed process once the solution is implemented. Given the limited time available for development, the project's focus is on delivering the application's core and essential features. ASP.NET Core MVC 5 web application, .NET Core Web API, and a SQL Server database are a few of the main outputs. The most recent version of .NET will be used to create this project.

To ensure efficient and successful development, the application will be developed using agile methodologies. Due to the limited time, the developers will concentrate on delivering the application's core functionality within the tight capstone deadline, giving a priority on a user-friendly design and an improved online jewelry rental system.

# Project Scope

The scope of the project is covering the Jewelry Rental Appointment and Admin application (web) requirements for the Online Jewelry Rental System.

## In Scope

The following areas are in the scope for this project:

Web Services

* Login/Logout
  + Identity Framework
* Landing Page
  + Viewing of Jewelries
* Appointment Setting
  + Create an Appointment
  + Update an Appointment
  + Cancel an Appointment
* Manage User
  + Create User
  + Update User
  + Delete User
  + List User

Admin Application(Web)

* Login/Logout
  + View Profile
  + Change password
* Dashboard
  + Viewing of Jewelries
* Manage Users
  + CRUD
  + Pagination
  + Search
* Manage User
  + Create User
  + Update User
  + Delete User
  + List User

## Out of Scope

The following areas are out of scope for this project:

* The system does not include payroll, accounting, marketing, sales and other management systems.
* The customers do not have any access to the database and creation of admin and employees accounts.

# Current Process

The system's user-interface design and part of its necessary functions are currently being developed. The following tasks still need to be finished: setting up customer schedules, adding items to cart and the procedure involved, handling customer transactions and payments, deploying Kubernetes, and docker containerization.

# Functional Requirements

## Priority

The requirements in this document are divided into the following categories:

Security Officer Application

| **Item** | **Rating** | **Description** |
| --- | --- | --- |
| Login/Logout | High | User will be able to login with his provisioned account details. |
| Dashboard for Jewelries | High | Dashboard screens list of the jewelries. |
| List Users | High | List of Users which can be used for managing the CRUD operations |
| List Appointments and orders | High | Allows Users to create appointments and manage the orders  . |
| Profile | High | View/Edit user profile |

Rest API Online Jewelry Rental System

| **Item** | Rating | Description |
| --- | --- | --- |
| Login/Logout | High | User will be able to login with his provisioned account details. |
| Access Manage Users to perform CRUD operations | High | API’s to access the Users and perform CRUD operations |

Non-Functional Requirements

| **Item** | Rating | Description |
| --- | --- | --- |
| External 3rd party Application API | Low | Allow todo application to expose api’s specific to 3rd party tools to consume and list todos. |

# Glossary

*This section explains all of the terms and abbreviations that were used in this document, for those who are unfamiliar with them. Not everybody who reads this document will understand all of the terms, so this section is helpful.*

| **Term** | **Explanation** |
| --- | --- |
| Application | An act of putting something to use. |
| Software Application | is a computer program designed to carry out a specific task other than one relating to the operation of the computer itself, typically to be used by end-users. |
| Software | is a set of instructions, data, or programs used to operate a computer and execute specific tasks. |
| API | An application programming interface (API) is a way for two or more computer programs to communicate with each other. It is a type of software interface, offering a service to other pieces of software. |
| CRUD | is an acronym that comes from the world of computer programming and refers to the four functions that are considered necessary to implement a persistent storage application: create, read, update and delete. |
| Pagination | the sequence of numbers assigned to pages in a book or periodical. |
| ASP.NET Core | is a cross-platform, open-source developer platform for building many kinds of applications. |
| ASP.NET Core | is a web-development framework for building web apps on the .NET platform. |
| ASP.NET Core Identity | is an API that supports user interface (UI) login functionality. Manages users, passwords, profile data, roles, claims, tokens, email confirmation, and more. |

# References

*This section contains links to all other places that were referred to in this document. These may include:*

* *Documents on shared document control systems (such as SharePoint)*
* *Web sites*
* *URLs or network locations*

| **Name** | **Link** |
| --- | --- |
| ASP.NET Core Identity | [Introduction to Identity on ASP.NET Core | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-7.0&tabs=visual-studio) |
| REST API | [Tutorial: Create a web API with ASP.NET Core | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-7.0&tabs=visual-studio) |
| MSSQL | [What is MSSQL? About Microsoft SQL Server | Atlantic.Net](https://www.atlantic.net/vps-hosting/what-is-mssql/) |
|  |  |
|  |  |

# Appendix

*This section may include any other information that does not fit in the document above. This may include:*

* *Details of the processes*
* *Analysis of existing process and benefits for the Business Drivers section.*
* *Any other information you might need*

*Beware of overloading this document with information. Preferably, any extra documents should be created separately and linked to, rather than embedding, as it is better for file sizes and version control.*

# Document History

*This section details the history of the document at each version. It’s good to know what has changed in each version, by who, and when it happened.*

| **Version** | **Date** | **Changes** | **Author** |
| --- | --- | --- | --- |
| 0.1 | 03/03/2023 |  | Reddy |
| 0.3 |  |  |  |
| 0.4 |  |  |  |
| 0.5 |  |  |  |
|  |  |  |  |