Software Requirements Specification

for

SwenCo Online Comic Book Store

Version *1.0*

Prepared by Debashis Jena

UtsavTech, LLC.

03/26/2021

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3. System Features 5

3.1 Identify and notify arson events 5

3.2 Display people with high body temperature who are possibly COVID infected 5

3.3 Live online auction of nonfiction merchandise 5

3.4 Implement billing system for the auction 5

3.5 Enable online users to control the drone cameras virtually 5

3.6 Identify non-violent group by the arm movement 5

4. External Interface Requirements 6

4.1 User Interfaces Overview 6

4.2 Hardware Interfaces 6

4.3 Software Interfaces 6

5. System Features/Modules 7

5.1 Identify and notify arson events 7

5.2 Display people with high body temperature who are possibly COVID infected 8

5.3 Live online auction of nonfiction merchandise 9

5.4 Implement billing system for the auction 10

5.5 Enable online users to control the drone cameras virtually 11

5.6 Identify non-violent group by the arm movement 12

6. Nonfunctional Requirements 13

6.1 Performance 13

6.2 Security 13

6.3 Quality 13

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| *Debashis Jena* | *03/31/2021* | *Initial version* | *1.0* |
|  |  |  |  |

# Introduction

## Purpose

This SRS document describes the requirements for the release 1.0 of the online comic bookstore (OCB). This document covers both functional and non-functional requirements for this release. This is the primary document that needs to be referred by the development and the QA team for implementation and validation purposes. Each of these requirements is a high priority for release 1.0.

## Document Conventions

This document is created for the SwenCo Comic Book Shop to set up an online bookstore that should have the ability to auction the books and figurines, virtually control the drones that are in the store to view the merchandise, notify Captain Maryland if anyone is detected with arson or infected by COVID. Each of the high-level requirements mentioned in section 2 will be detailed in subsequent sections.

## Intended Audience and Reading Suggestions

The SRS document has been created for the entire product team, which includes the project manager, analysts, developers, and testers. It is recommended that this document is read in a specific sequence. It can be started with the introduction section which gives a brief idea about the project purpose and scope. The section that follows the introduction section is the overall description of the project and lists out the most important features of the application. Section 3, 4, and 5 describe the requirement specification in detail. So, the reading is suggested to be from the top to bottom.

## Product Scope

This phase of the project is aimed towards having features like trading comic books, non-comic-book merchandise like figurines, gather data about a crowd to determine if it is a violent or non-violent group, location tracker using the drones. The details of the features will be elaborated on in the subsequent sections.

## References

For this document, many sources were referenced. Here is a list of the sources which includes the task that required the author to create this document.

1. Davis, B. (2013). Mastering Software Project Requirements: A Framework for Successful Planning, Development & Alignment (1st ed.). (Ch 11) Plantation FL: J Ross Publishing.
2. No Author. (n.d.). IEEE 830 template 2. Retrieved from <https://learn.umgc.edu/d2l/le/content/545040/viewContent/21223671/View>.
3. Linker, S. (n.d.). Instructions for the Final Project. Retrieved from https://learn.umgc.edu/d2l/le/content/545040/viewContent/21227581/View.
4. Linker, S. (n.d.). Bidders' Conference. Retrieved from https://learn.umgc.edu/d2l/le/content/545040/viewContent/21227634/View.

# Overall Description

## Product Perspective

The main objectives of Captain Maryland are to stop crime, notify the people with fever and organize an auction for the non-fiction merchandise in the comic store. The expectation is to have a comic bookstore webpage for SwenCo, which will have the features to display an arson event or the people with high body temperature fever, with the help of the satellites. The webpage should also contain the section to trade rare merchandise and an online auction for the merchandise. The system needs to have the capability to authenticate the registered users. The billing system will capture the payment method information at the time of signup, which will be required while bidding.

The following figure displays the context diagram for the instant message application.

## Product Features

The following list is the features that are going to be implemented in this project.

* Identify and notify arson events
* Display people with high body temperature who are possibly COVID infected
* Live online auction of nonfiction merchandise
* Implement a billing system for the auction
* Enable online users to control the drone cameras virtually
* Identify non-violent group by the arm movement

Diagram

Description automatically generated

## User Classes and Characteristics

|  |  |
| --- | --- |
| 1. Captain Maryland | Proprietor of SwenCo comics store |
| 1. Customers | Users who access the website to view the merchandise and bid in the auction |
| 1. Website | The webpage with the ability to conduct an auction and control the drones |
| 1. Turtles and the control system | The satellites owned by Captain Maryland monitoring the arson events and high body temperature people |
| 1. Drones | Drones in the store to view the merchandise |

## Operating Environment

|  |  |
| --- | --- |
| OE-1 | The SwenCo comic store website will run on the following browsers: Chrome 79 and above, Safari version 12.0 and above, Firefox 68 and above, Internet Explorer version 11, and Edge. |
| OE-3 | The application is going to be available as a website that can be open on mobile devices. |
| OE-4 | This application will be hosted in the AWS cloud server. The latest versions of EC2 servers will be used for better performance. |

## Design and Implementation Constraints

|  |  |
| --- | --- |
| CO-1 | For better maintenance, the modern microservices architecture type will be used for designing the whole application. |
| CO-2 | Being microservices and to have the best performance, NoSql databases like MongoDB or AWS’s native DynamoDb will be used for the backend database. |
| CO-3 | For the frontend user interface, the system will use Angular 7 and Bootstrap. |
| CO-4 | The backend platform for this application will be NodeJs. AWS being the main backend service provider, recommends using the NodeJs version 12, for building APIs and backend controllers. |
| CO-5 | For the Continuous Integration and Continuous Delivery (CI/CD) process, Jenkins and AWS’s CodePipeline will be used. |

## User Documentation

|  |  |
| --- | --- |
| UD-1 | Confluence pages will be created with the documentation of the use of the features. |
| UD-2 | A help page will be available on the application which has cross-linked HTML content to the confluence pages. |
| UD-3 | Videos explaining the use of each of the basic features will be available on some of those confluence pages. |

## Assumptions and Dependencies

|  |  |
| --- | --- |
| AS-1 | The project sponsor approves the budget for the new implementations. |
| AS-2 | The developing company has skilled developers to write programs for the new features of the application. |
| DE-1 | High-speed internet is available at the user’s location. Since this application is hosted on the cloud, the user is required to have an internet connection. |
| DE-2 | Drones are managed by the store manager; to recharge and ensure they are in working condition during operation hours. |
| DE-3 | The turtles are in working condition and a constant stream of data is being transmitted. |
| DE-4 | Required hardware gadgets are available with Captain Maryland to view the people with high temperatures. |

# System Features

This section gives a brief introduction to some of the system features. Section 4 and 5 will elaborate on the below system features.

## Identify and notify arson events

As the satellites detect any arson event, it would send a notification to Captain Maryland.

## Display people with high body temperature who are possibly COVID infected

Monitor people's body temperature and mark them as COVID infected if the temperature is higher than 100.4ºF.

## Live online auction of nonfiction merchandise

The comics store website should have a section for the merchandise auction. It should use a streaming service like YouTube or Brightcove.

## Implement a billing system for the auction

The website should display the upcoming auction events on the website. The website should have a section to allow users to register to participate in the auction. As part of the registration, the application must enforce the users to enter the billing-related information, like credit card or PayPal credentials.

## Enable online users to control the drone cameras virtually

The website must have an integration with the drone system. Since the number of drones is limited, the users need to take turns using the drones. Drones are only allowed to be controlled within the store.

## Identify non-violent group by the arm movement

As Captain Maryland would like to avoid any involvement of police or any imprisonment for the non-violent groups, turtles need to have the capability to use a direct wave to the areas with big crowds and have people get dispersed.

# External Interface Requirements

## User Interfaces Overview

|  |  |
| --- | --- |
| UI-1 | The user interface is going to be based on the Single Page Application (SPA) architecture. |
| UI-2 | The pages will not refresh upon state changes. |
| UI-3 | The help page will be available on each page of the application. |

## Hardware Interfaces

|  |  |
| --- | --- |
| UI-1 | Drones will be used by the users virtually. There will not be any hardware interface required by the users to view the camera videos. However, there will be an integration between the website server and the drone services. |
| UI-2 | The Turtles data will be streamed to the server hosting the database. No hardware interface is required by the users. |

The hardware interfaces involved in the drone and the satellites are out of the scope of this project.

## Software Interfaces

There will be multiple software interfaces required for this system.

### Website Auction interface (BrightCove)

|  |  |
| --- | --- |
| SI-1.1 | The auction video will be live-streamed. The platform used here will be BrightCove. |
| SI-1.2 | The current ASK price will be displayed at the top right corner. |
| SI-1.3 | The latest bidding price will also be displayed as the bidding commences. |

### Satellite (The Turtles) streaming view page

|  |  |
| --- | --- |
| SI-2.1 | The interface displaying the satellite data will be displayed on a webpage. |
| SI-2.2 | This page should display the arson events in a map view. |
| SI-2.3 | Captain Maryland should have access to this page apart from the SMS or push notification. |
| SI-2.4 | The streaming data should be stored in a NoSql database for a high-performance ability. |

### Drone control page

|  |  |
| --- | --- |
| SI-3.1 | A page should be available for the users to control the drones virtually. |
| SI-3.2 | With a limited number of drones, the application should display the number of available drones to be used. In case of the unavailability of drones, the system should display the number in a queue for all the requesting users. |

# System Features/Modules

## Identify and notify arson events

### Description and Priority

As the arson events occur the satellites will identify the event by detecting high temperature. Then the satellites will transmit a notification to the system. The event-driven application would trigger an SMS to Captain Maryland. Priority = High.

### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus: | The satellites (The Turtles) send notifications about arson events. |
| Response: | The server that listens to these notifications, gets triggered. |
|  |  |
| Stimulus: | The server runs the algorithms to find out the location of the event and the time. |
| Response: | Then the system sends a notification to ExactTarget to send an SMS to Captain Maryland. |
|  |  |
| Stimulus: | The fire is persistent even after the SMS was sent. This may happen in case of an issue with the SMS sends. |
| Response: | The system sends another SMS to Captain Maryland. |
|  |  |
| Stimulus: | The first responders are already at the arson event or have responded to the radio messages. |
| Response: | The system will not send any notification to Captain Maryland. |

### Functional Requirements

|  |  |
| --- | --- |
| REQ-1.1: | Upon an arson event, the turtles detect the temperature and send a notification to the system. |
| REQ-1.2: | As the notification is received, the system computes where the event has occurred. |
| REQ-1.3: | The system also verifies if the first responders have already arrived or responded to the arson incident. |
| REQ-1.4: | If the first responders have already arrived at the location or they have already responded to the message, the system does not do anything. |
| REQ-1.5: | If the system detects that the event has not been identified or responded to by anybody, then it sends a notification to Captain Maryland. |
| REQ-1.6: | If the system identifies that the Turtles have been sending the notification even after the first SMS was sent, it will send another notification to Captain Maryland. |
|  |  |

## Display people with high body temperature who are possibly COVID infected

### Description and Priority

The Turtles monitors all the people with high temperatures. As it detects people with a temperature of more than 100.4ºF, it sends a high-fever notification to the system. The “turtles” then mark those people with a green circle using their laser beams, which would warn the people around the sick ones to keep a safe distance from them. Priority = High.

### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus: | The “turtles” monitor the people and detect high temperatures of more than 100.4ºF. |
| Response: | The “turtles” first use their laser to draw a green circle around the people and then send a notification to the system with detailed information about the people. |
|  |  |
| Stimulus: | The system receives the notification with detailed information about the people. |
| Response: | Upon the receipt of the notification, the system gathers the data and saves the people's data in a database with infected people. |
|  |  |
| Stimulus: | The user interface reads the data from the infected people database. |
| Response: | Each of the people found in the database will be marked with an icon or a circle to notify people around to keep a safe social distance. |

### Functional Requirements

|  |  |
| --- | --- |
| REQ-2.1: | When the satellites identify the people with high temperatures it sends a notification with detailed information about the people. |
| REQ-2.2: | As the information is received at the server end, the system processes the information and reads the temperature from the data. |
| REQ-2.3: | If the temperature reads more than 100.4ºF, then it saves the data in a separate database table with details of the person. |
| REQ-2.4: | The API should read the infected people database and should return with the information about them. |
| REQ-2.5: | The user interface should display the infected people with an icon to warn the other people around them to keep a safe social distance. |

## Live online auction of nonfiction merchandise

### Description and Priority

Captain Maryland should be able to conduct an online auction using this platform. Priority = High.

### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus: | SwenCo Comic bookstore schedule an online auction ahead of time. |
| Response: | The system should post the details of the auction on the website. |
|  |  |
| Stimulus: | Customers register and join the auction on the auction day. |
| Response: | The system should live stream the video on the website using BrighCove or Discord platform. |
|  |  |
| Stimulus: | Captain Maryland's sidekick, Sandy would be the moderator for the auction and posts the auction item on the system with a base ASK price. |
| Response: | The auction items will be displayed on the auction page with the ASK price. |
|  |  |
| Stimulus: | The registered users select the items displayed with a bidding price textbox. |
| Response: | The entered amount is validated to ensure it is not less than the asking price or the latest bidding price. |
|  |  |
| Stimulus: | If there is a higher bidding amount entered, the amount entered by the other users becomes invalid. |
| Response: | Other users should be able to enter a higher amount for the auction. |
|  |  |
| Stimulus: | The bidding amount is accepted as the final bidding price. |
| Response: | The merchandise is marked as “Sold” and becomes unavailable for bidding any longer. |

### Functional Requirements

|  |  |
| --- | --- |
| REQ-3.1: | Sandy shall be able to start the auction with the live streaming platform Brightcove/Discord (TBD) on the auction day. |
| REQ-3.2: | As the user begins the auction, users should be able to view the live video of the merchandise being sold |
| REQ-3.3: | The system shall display the base ASK amount next to the merchandise. |
| REQ-3.4: | As the bidding starts, the highest bid amount starts displaying to all the logged-in users. |
| REQ-3.5: | As the users enter the bidding price, the system should validate if the amount is higher than the highest bidding price. |
| REQ-3.6: | When there is a timeout for merchandise or the moderator decides to close the auction for a specific item, the amount will be sold to the highest bidder. |
| REQ-3.7: | If the amount is not accepted, the system should still allow the users to place a higher bid. |
| REQ-3.8: | When merchandise is marked as "Sold", the system should make it unavailable for any further bids. |

## Implement a billing system for the auction

### Description and Priority

When the users sign up for the auction, they shall be required to enter a valid credit card. Priority = High.

### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus: | The moderator posts the advertisement about the upcoming auction event with the list of the merchandise on sale. |
| Response: | All the users visiting the site will be able to view the advertisement. |
|  |  |
| Stimulus: | The interested users click the auction link. |
| Response: | The system shall take the user to a login page with an option to log in for the existing users or register for the new users. |
|  |  |
| Stimulus: | The new users click the Register button. |
| Response: | The system should display the new form to enter the user detail and submit. |
|  |  |
| Stimulus: | The user enters the personal details with an email address, name, and shipping address and clicks the Next button. |
| Response: | The system should validate the entries and display the next page for payment method information asking for the credit card or PayPal information. |
|  |  |
| Stimulus: | The user enters the credit card details or the PayPal information and clicks submit button. |
| Response: | The payment method info is validated and preauthorized. |

### Functional Requirements

|  |  |
| --- | --- |
| REQ-4.1: | Upon the register button is clicked, the system should take the user to the login/register button. |
| REQ-4.2: | For an existing user, the login should take the user to the page to sign up for the auction directly. |
| REQ-4.3: | If the register as a new user button is clicked, the system should display the form to enter the details. If an existing user tries to register as a new user, the system should display the error that the user already exists. |
| REQ-4.4: | After the personal information has been accepted by the system, the system should require the user to enter the payment method information. |
| REQ-4.5: | If the payment method is not validated successfully, then the system displays an error saying the payment method information is invalid. |
| REQ-4.6: | The system shall use integration with a billing system like Stripe or Zuora to save the billing account information. |

## Enable online users to control the drone cameras virtually

### Description and Priority

This feature enables the users to virtually use the drones in the SwenCo store. Priority = Medium.

### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus: | The user logs in and navigates to the Virtual Cameras page. |
| Response: | The system shall display the available drones to the user. |
|  |  |
| Stimulus: | The user clicks on the “Request Drone View” button. |
| Response: | The system shall display an option to pick a drone from the available ones. |
|  |  |
| Stimulus: | In case of unavailability of the drones, the system requests to stay in the queue, by clicking the button "Add me to the Queue”. |
| Response: | The application shall display the user's number in the queue. |
|  |  |
| Stimulus: | The user uses the navigation keys to move the drone inside the store. |
| Response: | The system shall move the drone inside the store and the videos are live-streamed at the requesting customer’s view. |

### Functional Requirements

|  |  |
| --- | --- |
| REQ-5.1: | As part of the authenticated user’s view, there shall be a page to operate the drones. Without logging in, the page should display a button as “Log in to operate the drones”. |
| REQ-5.2: | The system shall display the number of available drones to operate. If all the drones are in use, the system shall display an option to add the user to a waiting list queue. |
| REQ-5.3: | A maximum time of 5 minutes is allowed for a drone use session. After 5 minutes, the drone should return to the dock and be available for the next use. |
| REQ-5.4: | As the users start operating the drones in the store, it should start a live video of the merchandise that the camera is focused on. |

## Identify non-violent group by the arm movement

### Description and Priority

Captain Maryland would like to identify the areas with big crowds, but with non-violent people. In such cases, she would like to avoid any involvement of police and any imprisonment. This feature will give the capability to identify such groups by their hand movement, using an AI technique. Priority = Medium.

### Stimulus/Response Sequences

|  |  |
| --- | --- |
| Stimulus: | A group of people gathers at a place. |
| Response: | The system shall identify the crowd by the number of people in the area. |
|  |  |
| Stimulus: | Non-violent people are part of the group. |
| Response: | The system shall detect the hand-movements of the people, presence of a weapon and determine if the group is a violent one or non-violent. |
|  |  |
| Stimulus: | The crowd is identified as a non-violent group. |
| Response: | The system within the "turtles" shall direct a wave at the crowd which will annoy them and causes them to disperse. |
|  |  |
| Stimulus: | The crowd is identified as a non-violent group. |
| Response: | The system within the "turtles" shall direct a wave at the crowd which will annoy them and causes them to disperse. The wave shall not cause any damage to anyone. |

### Functional Requirements

|  |  |
| --- | --- |
| REQ-6.1: | Upon a gathering of a big crowd, the system shall be able to identify the crowd. |
| REQ-6.2: | As the crowd is gathered, the system shall capture the data about it like:   1. Number of people 2. Min and Max temperature in the group 3. Average hand movement speed 4. Presence of weaponries; number by categories |
| REQ-6.3: | The data shall be sent to the “turtles”. |
| REQ-6.4: | If the group is persistent even after the wave is directed to them, the first responders will be contacted via radio or SMS. |

# Nonfunctional Requirements

## 6.1 Performance

NF-1.1: Being a public customer-facing web application, all the APIs are required to have a response time of fewer than 6 seconds.

NF-1.2: As the number of drones in the store is limited, the system should allow a slightly higher number of users requesting the use of them.

NF-1.3: The auction platform should be capable of allowing 1000 concurrent users during a live event.

## 6.2 Security

NF-2.1: Users must log in with named credentials, before accessing the features of the web application.

NF-2.2: As the application allows the use of credit cards, it should be PCI compliant.

NF-2.3: The network transactions shall use HTTPS protocol for communication.

NF-2.4: The software shall be hosted in a virtual private cloud.

NF-2.5: The log-in credentials shall be encrypted when data is at rest.

## 6.3 Quality

NF-3.1: The application is required to be available 99.99% between 6 am to midnight. The system can have an acceptable availability of 95% during the off hours.

NF-3.2: In case of server down, the system shall display a message conveying the same.