Project Name Here

MIS department, Tabor School of business, Millikin University |

Software Requirements and Specifications

Your Names Here

2024

Table of Contents

[Definitions 2](#_Toc174182736)

[Goals & Value 2](#_Toc174182737)

[Accuracy 2](#_Toc174182738)

[Increased Efficiency 2](#_Toc174182739)

[Other subsections 2](#_Toc174182740)

[Scope 2](#_Toc174182741)

[Timeline 2](#_Toc174182742)

[Use-Cases & User Stories (option 1) 3](#_Toc174182743)

[Data Flow (option 2) 3](#_Toc174182744)

[Non-Functional Requirements 3](#_Toc174182745)

[System Requirements and Functional Requirements 4](#_Toc174182746)

[OS and Development requirements 4](#_Toc174182747)

[User Interface Requirements 4](#_Toc174182748)

[Backend, Cloud, Network/External Connections, etc. Requirements 5](#_Toc174182749)

[Other Categories 5](#_Toc174182750)

[Risk, Security, and Other Considerations 5](#_Toc174182751)

[Works Cited 6](#_Toc174182752)

# Definitions

* **SDLC:** Acronym for Software Development Lifecycle which are the processes and project management models that organizations use to create software and information systems.
* Your version of AGILE/DEVOPS: <definition here>
* **OTHER TERMS:** <definition here>

# Goals & Value

Short paragraph or two explaining the goals of the company (its mission and/or value statement) and goals for the project and how they coincide with the criteria for success. To put it another way: Ensure you explain what the value of the project is to the company in this section.

Should have subsections here – for instance, if in paragraph 2 we stated that we were grading the project based on how it increases the accuracy or efficiency of operations:

## Accuracy

Describe the Criteria here – what does it mean to be accurate (or whatever subgoal is)? How does the project increase accuracy?

## Increased Efficiency

How are we measuring Efficiency? What are the metrics? How does the project increase this?

## Other subsections

Edit and/or remove these and add your own based on your project. Remember with these categories – real world examples work best. Don’t just say using Excel with automated functions for calculations will limit mistakes in a ledger. Take a screenshot of a good & bad example and explain them (*a picture is worth a 1000 words* and all that).

# Scope

How is this limited? We couldn’t fully build the software in the time allotted so there has to be at least that limitation and most likely a few others (everyone has a budget).

# Timeline

<Paste a timeline image (or 2 but not more than 2)> and describe the reason for the major milestone completion estimates.

I like to make these in PowerPoint, Visio, Draw.io, or other image/presentation software to make the timelines then just paste them here (using the estimates in GitHub). You can use any software that lets you generate a timeline – can use AI for this (not making the estimates or describing the reasoning behind those estimates but building the timeline from the dates you have estimated).

# Use-Cases & User Stories (option 1)

<INSERT USE-CASE DIAGRAM HERE>

Explain with use-case explanation or using Epics and User Stories below.

**IF DFD IS USED**: DELETE THIS SECTION. (Delete this line either way)

# Data Flow (option 2)

<INSERT HIGH LEVEL DATA FLOW DIAGRAM HERE>

Explain the overall flow of operations.

<INSERT LEVEL 2 DATA FLOW DIAGRAM HERE>

Explain the detailed flow of data here including how it will affect automation and the project.

**IF USE-CASE IS USED**: DELETE THIS SECTION. (Delete this line either way)

# Non-Functional Requirements

Include important Non-Functional Requirements here: those that are included in your GitHub backlog should also have a link to that backlog for more information.

Standard way is to insert them in a Table (make it in Excel…\*cough\* export tasks to csv as seen below) that you have formatted. Or as a series of bullet points (see [Functional Requirements](#_System_Requirements_and) for that example):

A screenshot of a computer

Description automatically generated

# System Requirements and Functional Requirements

See [Non-Functional Requirements](#_Non-Functional_Requirements) for image on how to download your project backlog.

Again, only include the important system and functional requirements then link to backlog for full amount (at least 20).

I will make this a little easier, as it can be hard to figure out if a requirement is a non-functional system requirement we should track or just a non-functional requirement. Here if it involves the system itself (infrastructure or technology used) or if it is an actual Functional Requirement (*The system shall…*). Should split up by category where possible:

## OS and Development requirements

* Due to the requirements of the client and the systems used by the company, the system will be built on the \_\_\_\_\_\_\_\_\_\_\_\_\_ OS and using the \_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_, and \_\_\_\_\_\_\_\_ languages.
* Add any others here (like what testing tool will you use or IDE or using GitHub <- as you are)
* Can also stipulate the minimum requirements for running the software as it connects to OS (like it should run on a browser or Cell phone).

## User Interface Requirements

* Something, something, screen

## Backend, Cloud, Network/External Connections, etc. Requirements

* There are many ways to split up the categories. (External Connections or Cloud can also be where you mention GitHub if OS was too big or you have a lot to put in Cloud/External)

## Other Categories

* This can also be a table but do not just have a “category” column. Sort/filter and copy and paste it by the category’s column so that this document’s table of contents links directly to various categories (for the c-suite, testers, and users who don’t have full access to the backlog or Excel spreadsheet).

# Risk, Security, and Other Considerations

Include any security considerations or other items which don’t fit in the main sections here. Should include some considerations on liability, regulations, information security, cyber security, risk management, or any other items that you discussed with sponsors that limit the scope or have to be integrated into planning and building the system.

# Works Cited

Graffius, S. (2016). *Agile Scrum: Your Quick Start Guide with step-by-step instructions.* North Charleston, SC: CreateSpace.