***Project Name*** Requirements Document

*To use this template:*

1. *Replace any blue italicized text with your own text. You may remove or add sections as needed for your particular projects.*
2. *Delete these instructions and any other italicized instructions.*
3. *Pay particular attention to the notes in RED.*
4. *Since this is customer-written, it may sound informal. That’s OK.*
5. *It should contain enough information to permit design.*
6. *Section 3 is very important. It is often a list.*

# *Team Number:*

# *Team members who participated:*

# *Team members who did NOT participate:*

# 1. Introduction

## 1.1 Overview

*Give a brief overview of what the project will accomplish. Is it a website to support fundraising for a nonprofit? Is it a mobile app to help you find your favorite taco place? This should be a few sentences (no more than a paragraph) about what the overall vision is. It should mention what problem you are trying to solve.*

## 1.2 Scope of the Product

*Include a brief narrative here which describes the product as you intend it to be realized. Use this section to define boundaries and set expectations (constraints, not design). For example, if this is a mobile app, will it support iOS and Android, or only one or the other? If it is a ML project, what frameworks can it be deployed on - spark, tensorflow, MX? If it is a web application, will it support an API? This should define at a high level what the boundaries of your deliverables are.*

## 1.3 Business Case for the Product

*Why is this product required? How will it contribute to the goals of your institution? This section can be used when requirements are being negotiated, to assess whether a particular change is a good idea. This section also helps readers understand why certain requirements have been included. In this section, pretend that you are the customer.*

# 2. General Description

*This section will give the reader an overview of the project, including why it was conceived, what it will do when complete, and the types of people we expect will use it. We also list constraints that were faced during development and assumptions we made about how we would proceed. Note that this section COULD NEVER be written by engineers.*

## 2.1 Product Perspective

*Why have you chosen to develop this product? What need does it serve? Who are the primary stakeholders, who is developing the project, and who will benefit from the finished product? In this section, you must delineate between customers and users. They may be the same, or they may not (eg. Therapists vs. Students)*

## 2.2 Product Functions

*What does your product do? What activities can users perform while using it? List the main functions that you will build into your product here. For example, if you were creating an application to generate exams, the main functions might be that it will allow the ability to enter, edit, and review questions, and that it can generate exams in different formats and lengths. The details will come in the specific requirements - this should cover the high level.*

## 2.3 User Characteristics

*Who do you expect to use your finished product, and why? What is their technical background, their training or education, their motivation to use it? What obstacles might they encounter, and what specialized skills will they need? For example, an application designed to let a security expert quickly assess the state of the network, firewalls, and intrusion detection logs will have a very different approach than an application designed to let elementary school students learn math by playing games. Sometimes, the reading grade level, the computer expertise level, the motors skill level, hearing & speaking level, etc. are portrayed here.*

## 2.4 General Constraints

*Did you work under any constraints such as platform or development environment? Did you have to make your product compatible with any existing software or other products currently in use? For example, if you are developing a piece of software that has to run on department severs, you may be limited to MySQL and PHP as your development environment, as they do not support frameworks like Laravel or Flask. The overall cost of the system for the customer AND FOR THE USER are included here, eg. it must work on a $100 tablet.*

## 2.5 Assumptions and Dependencies

*In this section, list any assumptions you made about your project (for example, did you assume that the finished product would need to be delivered over the internet? Do you assume that all users will have google or facebook accounts? Does it require that data be available as a starting point for analysis?). If your project depends on any particular technical infrastructure, or requires administrators or others with specific skills, note that here. This section is generated by the vagueness of the process.*

# 3. Specific Requirements

*This section of the document lists specific requirements. Requirements are divided into the following sections:*

1. *User requirements. These are requirements written from the point of view of end users, usually expressed in narrative form.*
2. *System requirements. These are detailed specifications describing the functions the system must be capable of doing.*
3. *Interface requirements. These are requirements about the user interface, which may be expressed as a list, as a narrative, or as images of screen mock-ups.*

## 3.1 User Requirements

*List user requirements (functional) here. This should be a numbered list of your commitments. What will the user be able to do? Login? Create an account? View their account balance? Send messages to other users? View detailed reporting? Set up email notifications? This is the heart of your contract with the stakeholders. These requirements should each define a user function, and should be testable by the user during UAT.*

## 3.2 System Requirements

*List detailed system requirements (non-functional) here. If your system is large, you may wish to break this into several subsections. System requirements are those that the system must support but that may not be testable by an individual user. For example, does the system need to log all transactions? Does it need to support two factor authentication? Do all reports need to render in 2 seconds or less? Does it need to be deployed by automated script? Does it need to replicate data to multiple databases? These usually operate in the background.*

## 3.3 Interface Requirements

*List interface requirements here; or* ***include wireframes / screen mockups****. If you use mockups, be sure to explain major features or functions with narrative to avoid confusion or omission of desired features. If there is a user interface, you MUST consider usability factors here - error handling, navigation, consistent presentation, etc.*

# 4. Appendices

*If you wish to append any documents, do so here. You may wish to include some or all of the following:*

· *~~Personas and scenarios developed for this project~~*

· *~~Transcripts of user interviews, observations, or focus groups~~*

· *Copies of communications which contain user requirements*

· *Original project proposals or other historical documents*

· *Lists of similar projects or products, with notes about how they differ from yours*

· *A list of requirements which were "wish-listed" or marked unfeasible at present*

· *Original screen mockups, if they are relevant*

# 5. Glossary

*Include a glossary of definitions, acronyms, and abbreviations that might be unfamiliar to some readers, especially technical terms that may not be understood by end-users or domain-specific terms that might not be familiar to developers. However, if you did a good job with an understandable narrative (above), this might be very short.*

# 6. References

*List references and source documents, if any, in this section.*