<Project Name>

Vision Document

Version: 1.0

# 1.                  Introduction

# 2.                  Positioning

## 2.1               Problem Statement

|  |  |
| --- | --- |
| The problem of | Time-consuming exam administration/grading and academic dishonesty |
| affects | Academic institutions/instructors with large class sizes |
| the impact of which is | A longer delay in exam performance feedback with fewer opportunities for examination and instructor exhaustion/burnout |
| a successful solution would be | An exam administration suite that generates random exam layouts for each student to prevent academic dishonesty, automates a large amount of the grading process, and creates personalized feedback on exam results. |

## 2.2               Product Position Statement

|  |  |
| --- | --- |
| For | Instructors |
| Who | Have large classes and are overwhelmed by the amount of work exam administration entails |
| The Handy Test Manager Thingy | is an exam creation and grading solution |
| That | Relieves the instructor’s worry of academic dishonesty while reducing the time involved with grading and providing feedback |
| Unlike | Xerox Remark Software |
| Our product | Provides an affordable, personalized solution. |

# 3.                  Stakeholder Descriptions

## 3.1               Stakeholder Summary

| **Name** | **Description** | **Responsibilities** |
| --- | --- | --- |
| Client | Clinton Rogers | The client will describe the intended usage of the software to our team. They will deliver timely responses to inquiries about the design of the software. The client will provide opportunities to test the software using at least one actual exam. The client will offer feedback on software performance whenever a functional build is provided for testing purposes. |

## 3.2               User Environment

[Detail the working environment of the target user. Here are some suggestions:

Number of people involved in completing the task? Is this changing?

How long is a task cycle? Amount of time spent in each activity? Is this changing?

Any unique environmental constraints: mobile, outdoors, in-flight, and so on?

Which system platforms are in use today? Future platforms?

What other applications are in use? Does your application need to integrate with them?

This is where extracts from the Business Model could be included to outline the task and roles involved, and so on.]

# 4.                  Product Overview

## 4.1               Product Perspective

Our product will be a self-contained solution, requiring the hardware support of any scanning device. It will be operated locally on the following operating systems: \_\_\_\_\_\_\_\_\_\_\_

[This subsection of the **Vision** document puts the product in perspective to other related products and the user’s environment. If the product is independent and totally self-contained, state it here. If the product is a component of a larger system, then this subsection needs to relate how these systems interact and needs to identify the relevant interfaces between the systems. One easy way to display the major components of the larger system, interconnections, and external interfaces is with a block diagram.]

## 4.2               Assumptions and Dependencies

[List each factor that affects the features stated in the **Vision** document. List assumptions that, if changed, will alter the **Vision** document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the **Vision** document will need to change.]

## 4.3               Needs and Features

|  |  |  |  |
| --- | --- | --- | --- |
| **Need** | **Priority** | **Features** | **Planned Release** |
| Random exam generation | 4 | Individualized exams for each student | February 2015 |
| Student list creation | 3 | Importing of lists of students from other sources, addition/deletion/modification of existing student roster | March 2015 |
| Multiple choice answering sheets | 4 | Each random exam will be mapped to an answer sheet using QR code references | February 2015 |
| Exam creation | 5 | Allow questions to be created, modified, deleted and grouped for examinations | January 2015 |
| Multiple choice response scoring | 5 | Allows answering sheets to be scored with heavy automation from the software, soliciting user feedback when necessary. | April 2015 |
| Score management | 2 | Record and track scores on a per-student basis. Allow instructor to override or scale results. | May 2015 |
| Extended formatting for questions and answers | 3 | Allows questions and answers to be formatted with a wider range of text options. Also allows for the usage of externally created diagrams and images. | March 2015 |
| Non-multiple choice assisted scoring | 3 | Allows the instructors to use software to manually grade and provide feedback on open-ended questions. | April 2015 |

## 4.4               Alternatives and Competition

[Identify alternatives the stakeholder perceives as available. These can include buying a competitor’s product, building a homegrown solution, or simply maintaining the status quo. List any known competitive choices that exist or may become available. Include the major strengths and weaknesses of each competitor as perceived by the stakeholder or end user.]

# 5.                  Other Product Requirements

The product will be required to score multiple-choice responses correctly with at least 99.9% accuracy. The storage of exam questions and individualized question orders must be persistent. A user manual will be provided upon the release of the software that will document core features of the software. We will be responsible for installing the software for the client.

[At a high level, list applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements.

Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.

Note any design constraints, external constraints, or other dependencies.

Define any specific documentation requirements, including user manuals, online help, installation, labeling, and packaging requirements.

Define the priority of these other product requirements. Include, if useful, attributes such as stability, benefit, effort, and risk.]