

---

# Software Requirements Specification

for

**<Project>**

Version 1.0 approved

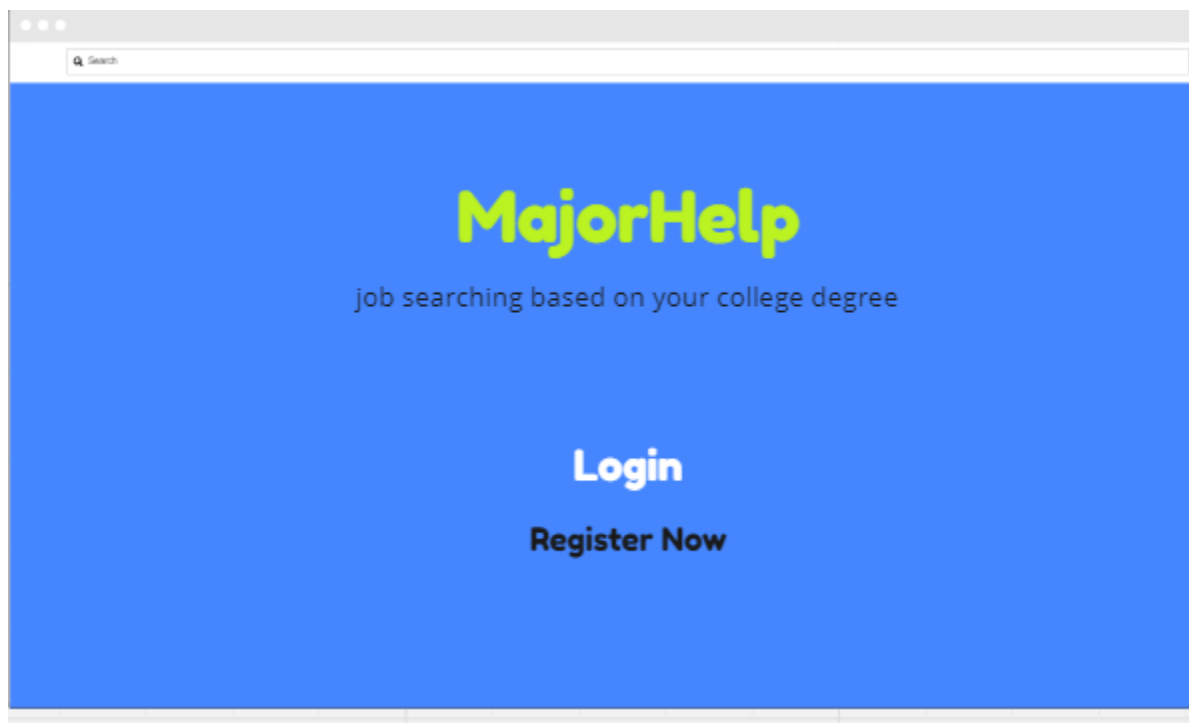
Prepared by <authors>

<organization>

<date created>

# Table of Contents

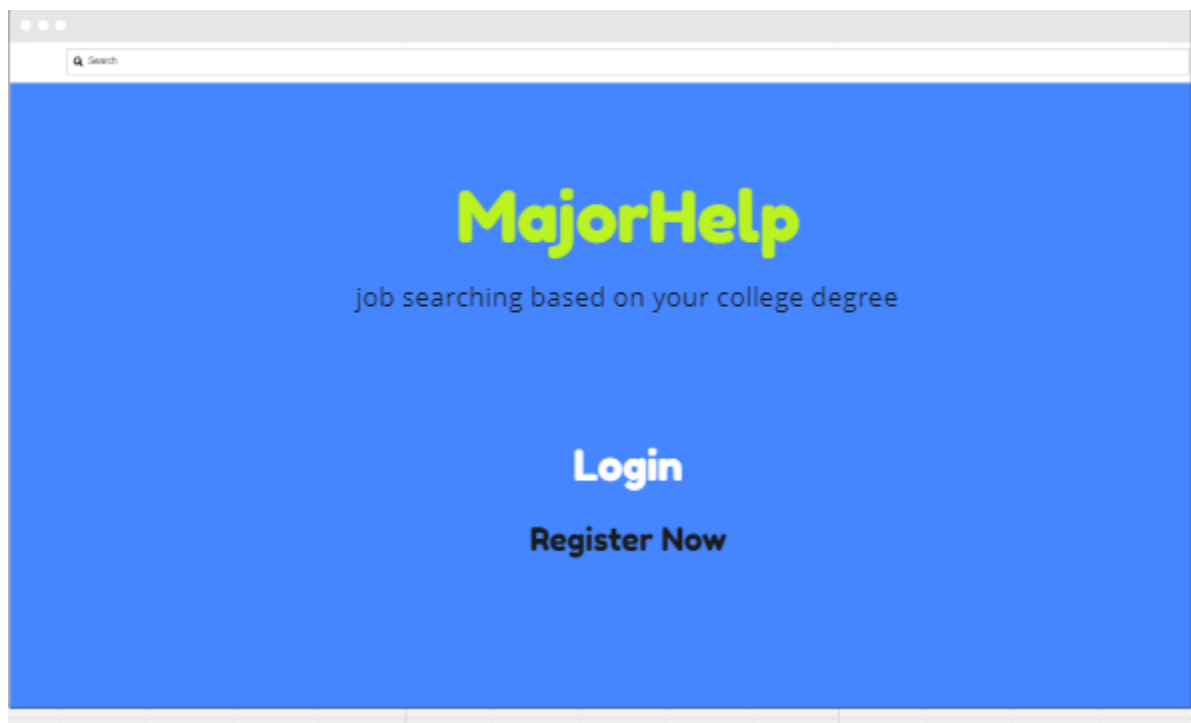
<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
<b>3. External Interface Requirements</b>	<b>3</b>
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
<b>4. System Features</b>	<b>4</b>
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
<b>5. Other Nonfunctional Requirements</b>	<b>4</b>
5.1 Performance Requirements	4



5.2	Safety Requirements	5
5.3	Security Requirements	5
5.4	Software Quality Attributes	5
5.5	Business Rules	5
<b>6.</b>	<b>Other Requirements</b>	<b>5</b>
<b>Appendix A: Glossary</b>		<b>5</b>
<b>Appendix B: Analysis Models</b>		<b>5</b>
<b>Appendix C: To Be Determined List</b>		<b>6</b>

## Revision History

Name	Date	Reason For Changes	Version



# 1. Introduction

## 1.1 Purpose

*<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>*

## 1.2 Document Conventions

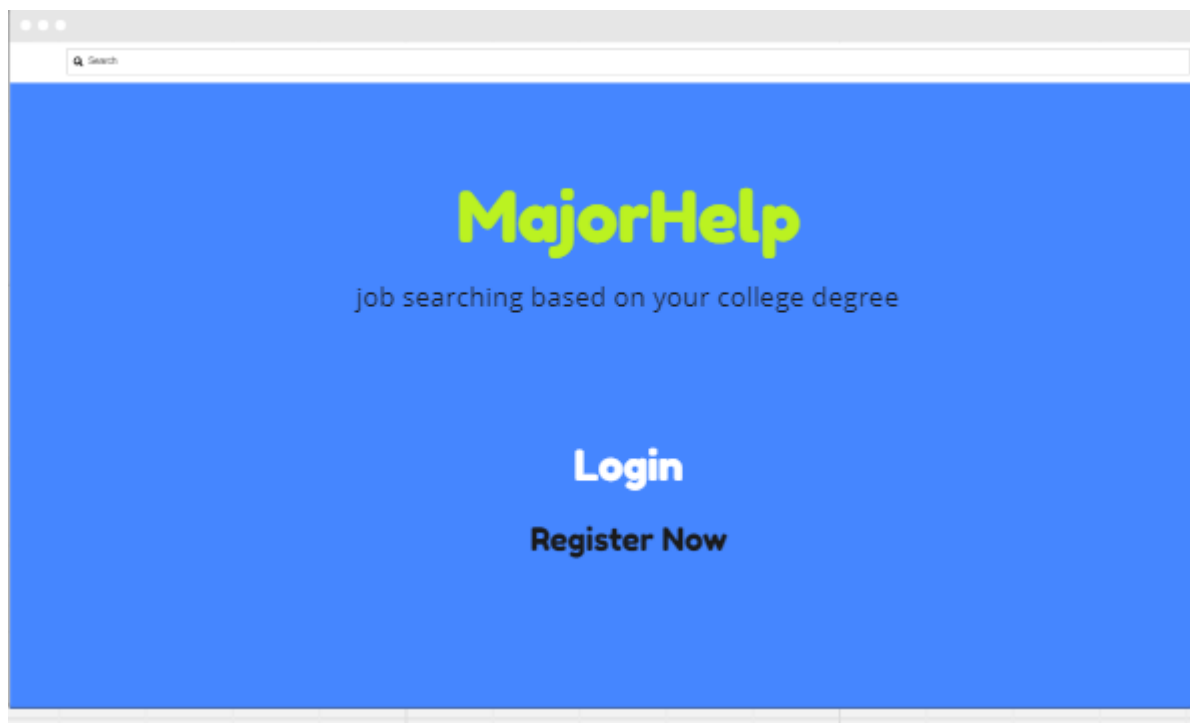
*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## 1.3 Intended Audience and Reading Suggestions

*<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>*

## 1.4 Product Scope

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a*



separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## 1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

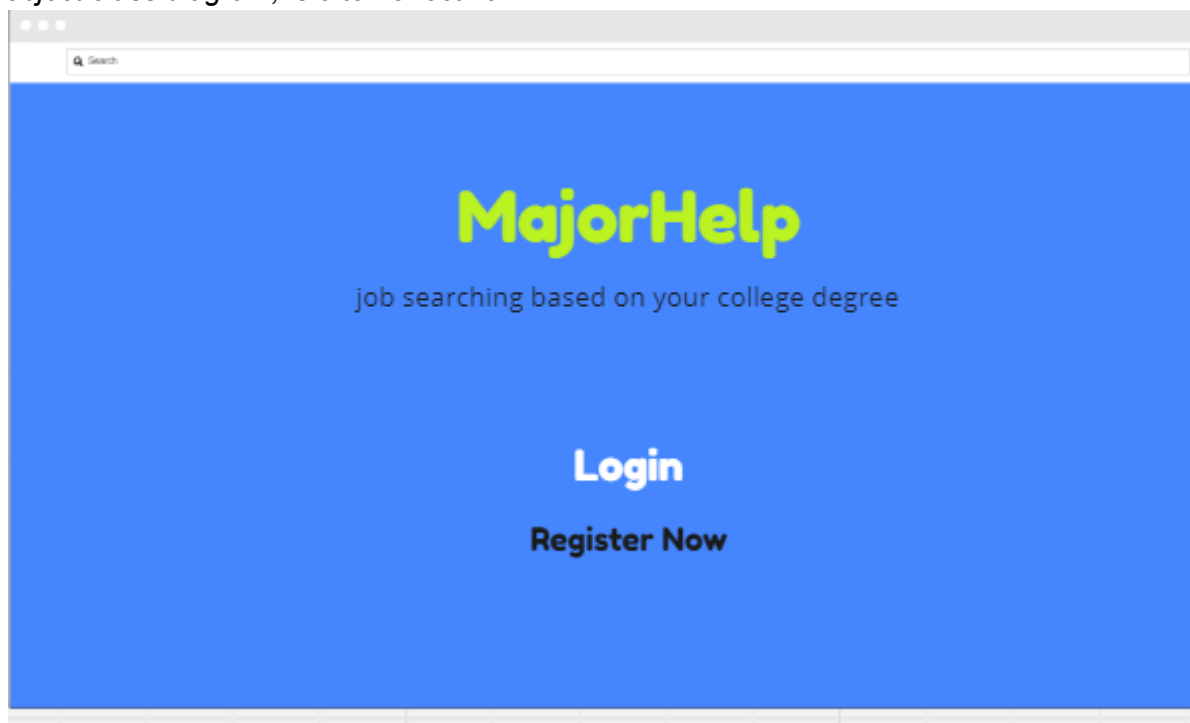
## 2. Overall Description

### 2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

### 2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>



## 2.3 User Classes and Characteristics

*<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>*

## 2.4 Operating Environment

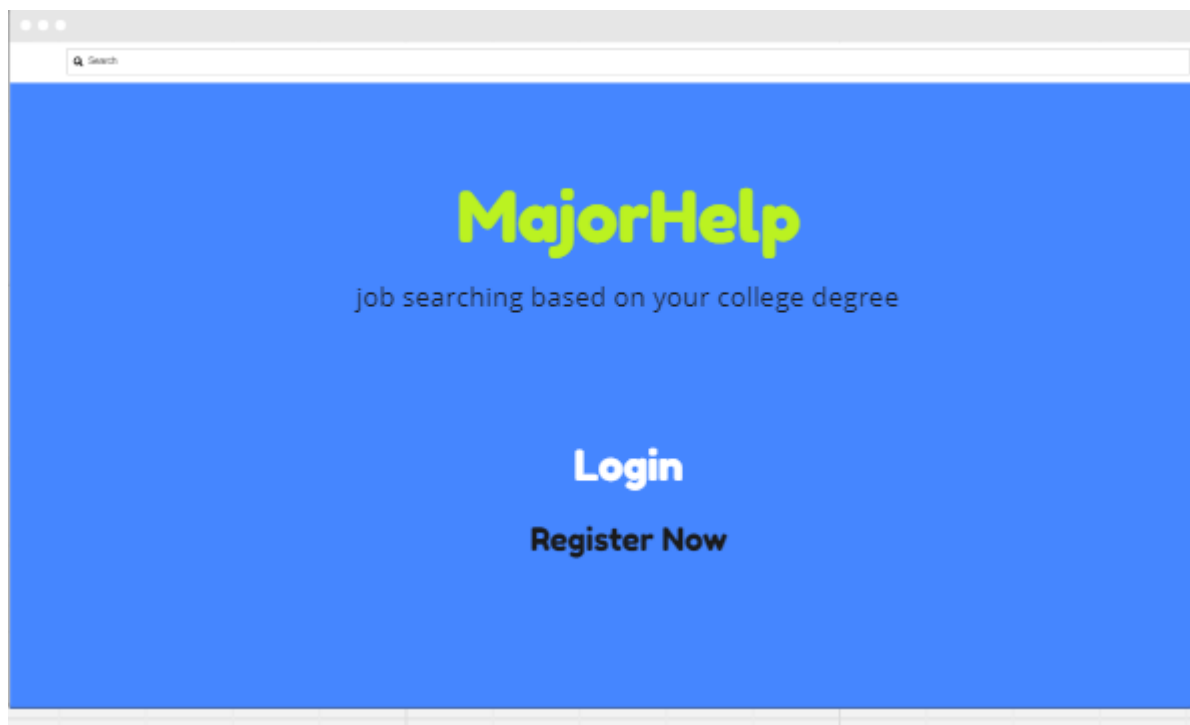
*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## 2.5 Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>*

## 2.6 User Documentation

*<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>*

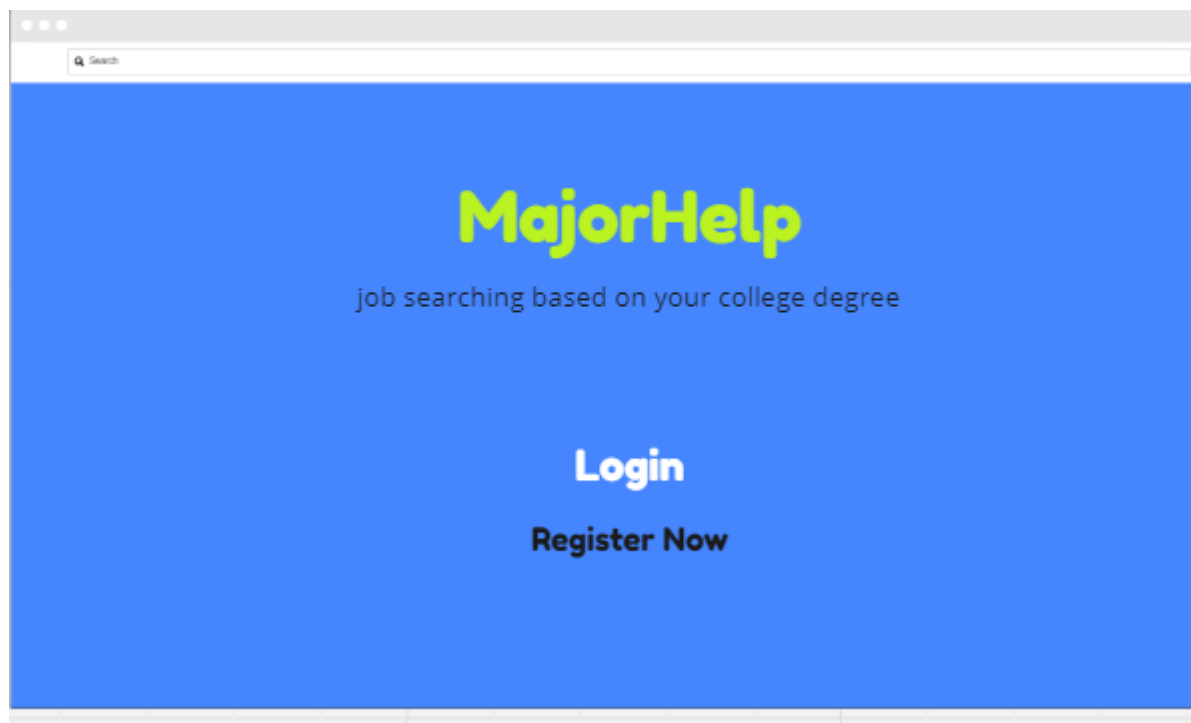


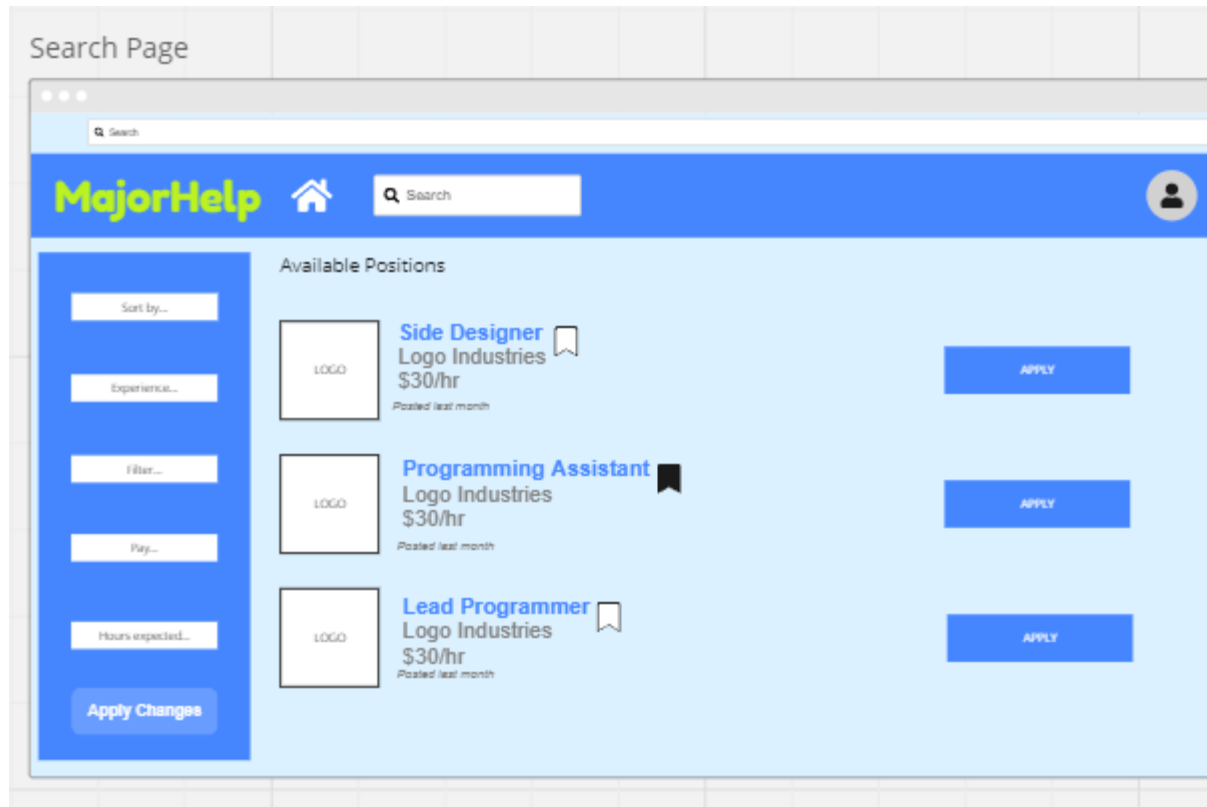
## 2.7 Assumptions and Dependencies

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

## 3. External Interface Requirements

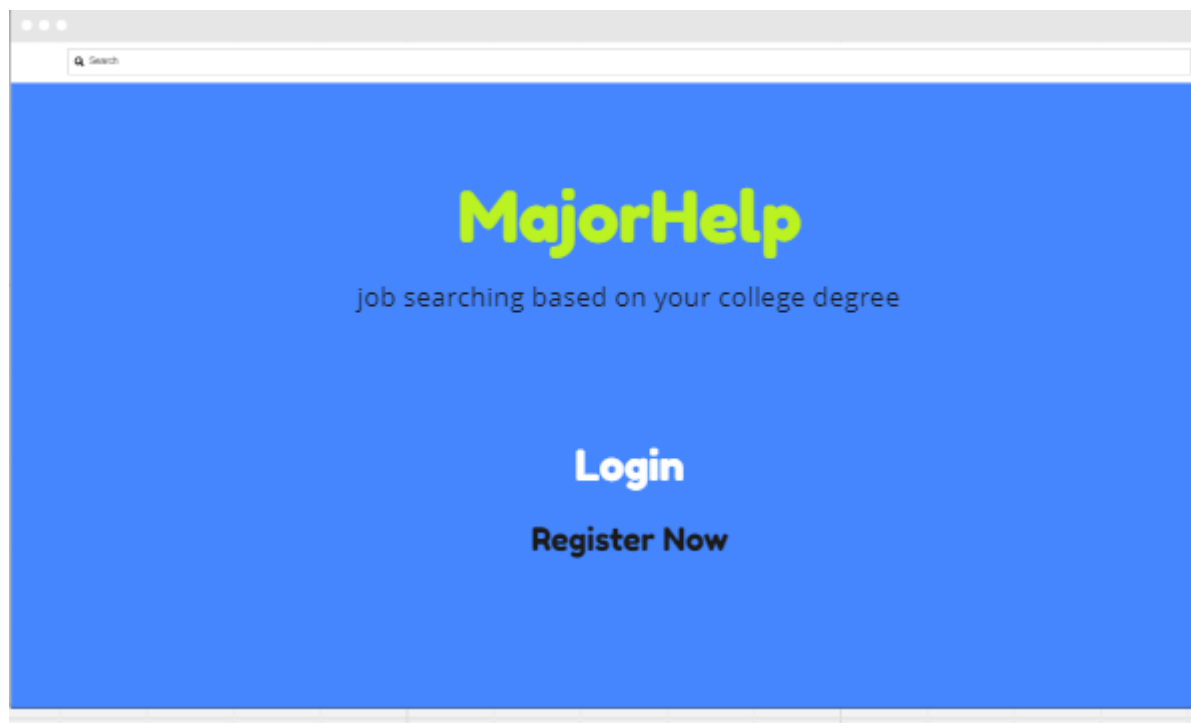
### 3.1 User Interfaces





## 3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>





### 3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

### 3.4 Communications Interfaces

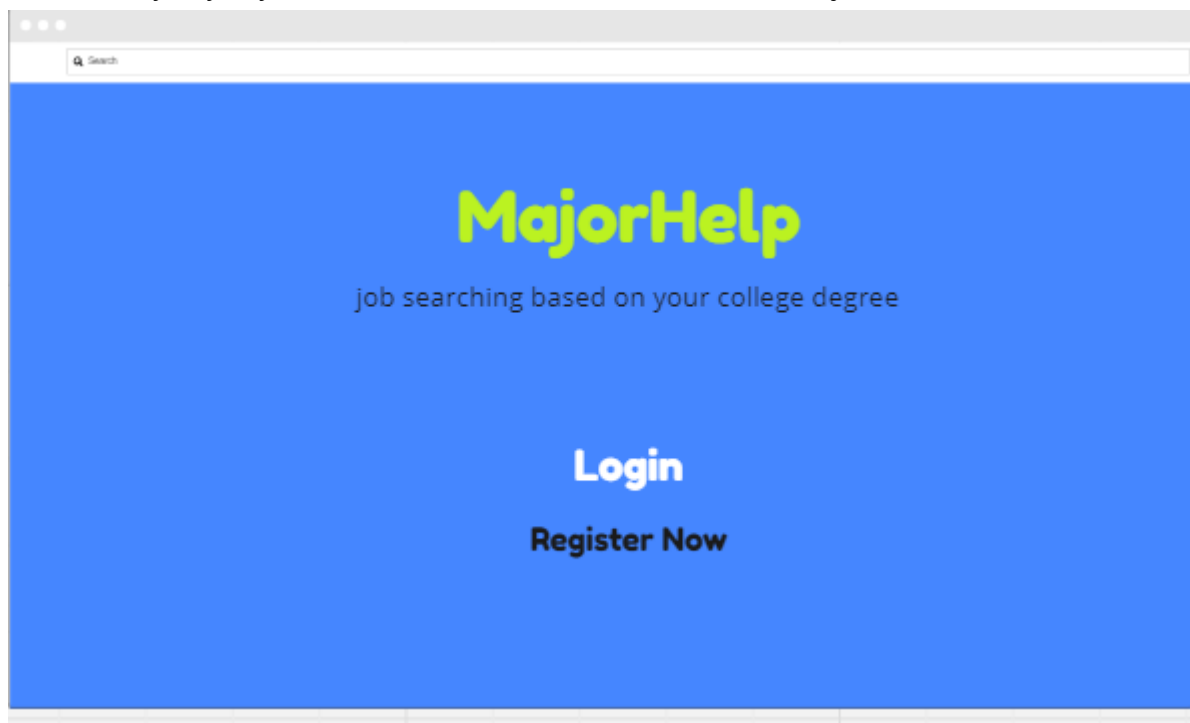
<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

## 4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

### 4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>



#### 4.1.1 Description and Priority

A job database would entail a place for job listings to be stored for later reference and use. This database would exist outside of the main application, and data would need to be retrieved by the application. This database will be storing the jobs found from other sites. It is of high priority to include on account of how important having a job aggregation database is to the functionality of this service— without it, there would be no content to load on the service.

The main **benefit** would be organization and efficiency, as the app would not be storing all of the database into itself, which would lead to increased loading times on the software.

**Costs** associated with this feature would be the amount of time required to get information from the database mentioned.

**Penalties**, therefore, come from whenever the database cannot be accessed through the server (whenever there is downtime); users end up being unable to view jobs when searched.

**Risk** associated with the database would be the same as its costs.

#### 4.1.2 Stimulus/Response Sequences

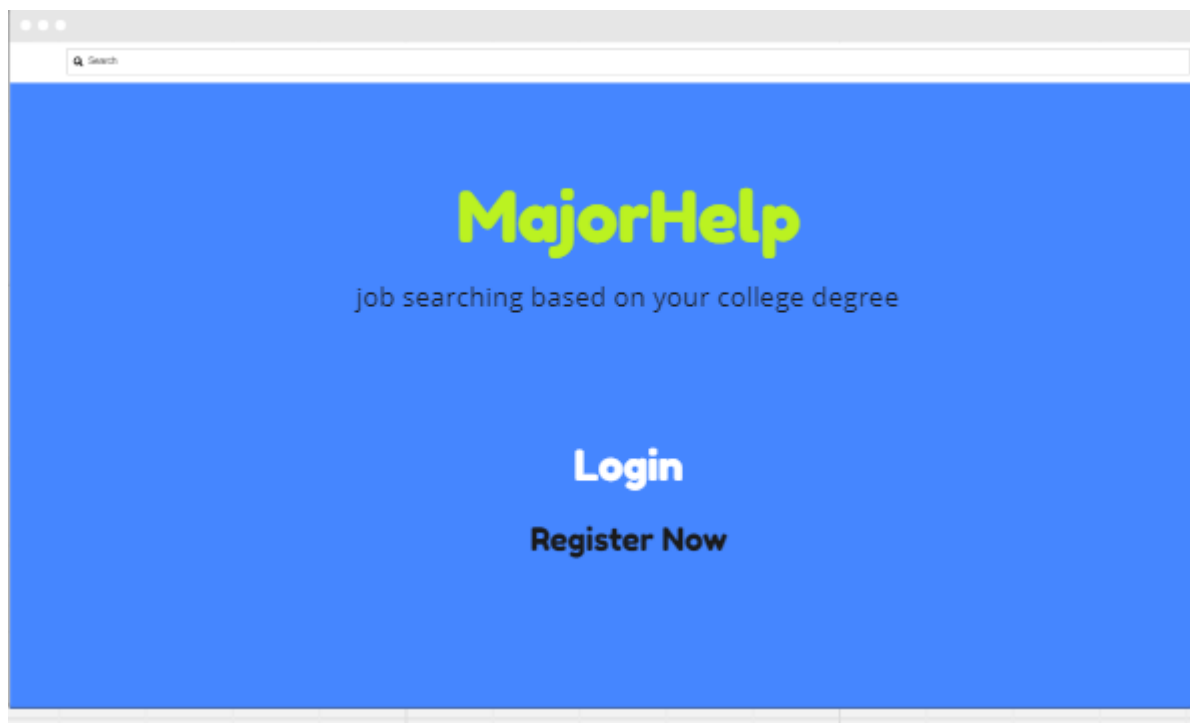
When a user searches for job listings, the system will call upon jobs from said database through a function and return:

1. If a job is found in your location, the site will display listings *based on recency* by default.
2. If no jobs are found due to *geographic location*, the site will return “No job listings found in your area! Change radius of search?”
3. If no jobs are found due to *error accessing the database*, the site will return “Error loading jobs. Try refreshing.”

The database (WIP)

#### 4.1.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the*



*services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1: *The system must allow the user to see where jobs are located*

REQ-2: *The system must allow an employer to show what their company is about*

REQ-3: *The system must filter out jobs using the filter feature*

REQ-4: *The system must show employers job listings*

REQ-5: *The system must show a users skill listings*

REQ-6: *The system must provide a privacy setting*

REQ-7: *The system must not show unverified users*

REQ-8: *The system must verify the user is human*

REQ-9: *The system must give the option for an authenticator*

REQ-10: *The system will display a list of work opportunities that are relevant to current college students.*

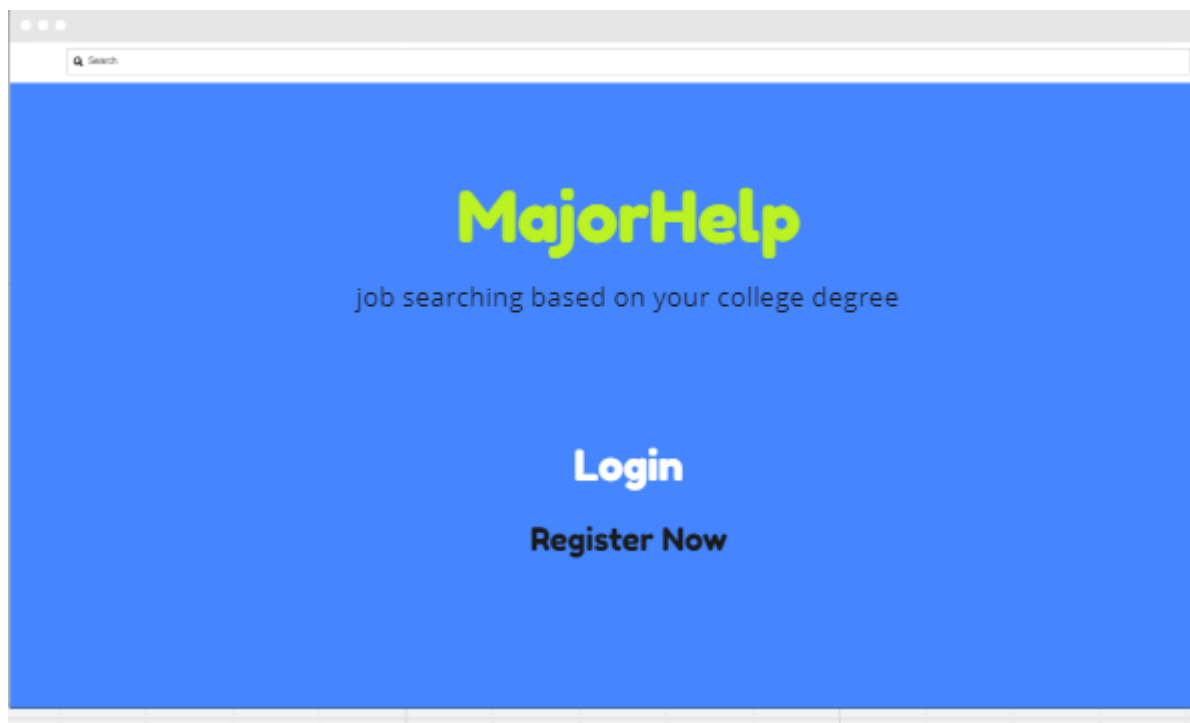
REQ-11: *The system will have an option to filter work opportunities by <some criteria>.*

REQ-12: *The system will have an option to sort work opportunities by date posted, expected pay, and distance.*

REQ-13: *The system will provide links to full listings and more information for a specified work opportunity when a user requests it.*

REQ-14: *The system will show a user the risks associated with that job*

REQ-15: *The system will scrape for job offerings for the user*



## 4.2 System Feature 2 (and so on)

### Application process

#### 4.2.1 Description and Priority

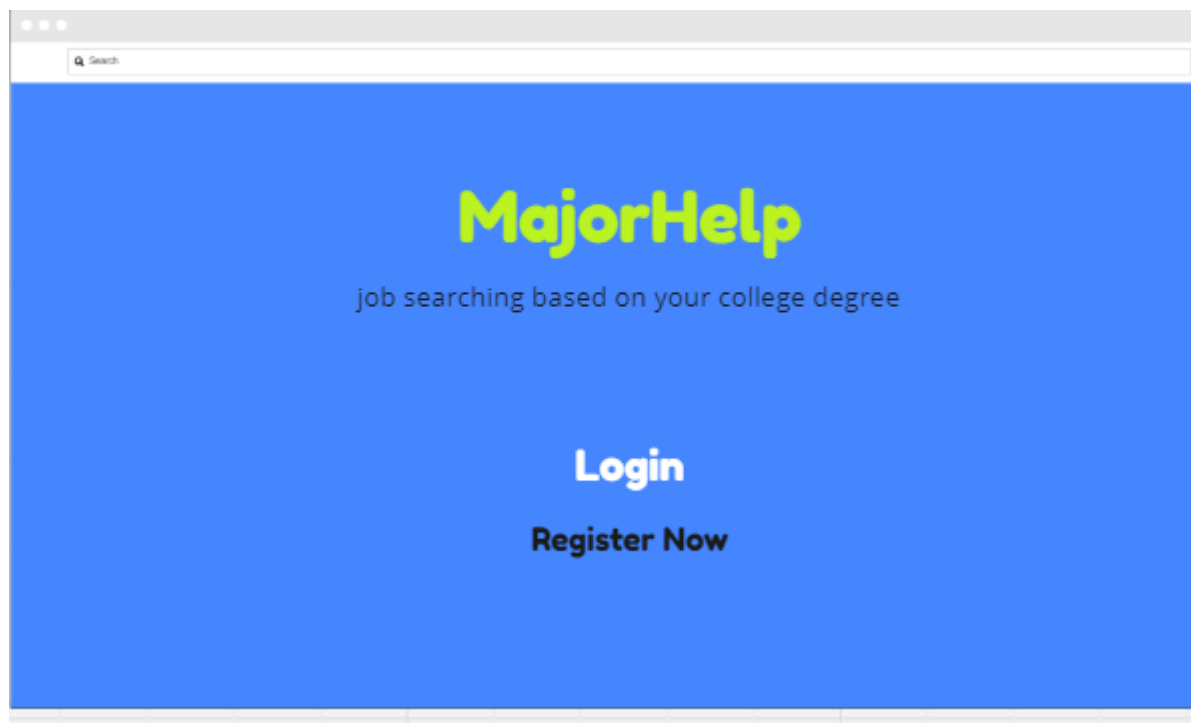
The application process for this service should be simple and straightforward. The application feature on the service will be in the form of a button on job listings that allows a user to

#### 4.2.2 Stimulus/Response Sequences

When a user logs into the service, the database should check the username and password given to it to see if it matches the correct username and password of the user. In terms of response, the database should let the user know if their information given was validated or not:

1. If the user logs in with the correct username and password, the service should give them access to the job listings/whatever the home screen should be.
2. If the user logs in with an incorrect password or username, the service should make note that "Password and/or username is invalid. Please try again or try to reset your password."

If a user is making a new account, then prompts associated with registering should be made available to make it abundantly clear if an attempt to join the service has been made possible. Different fields being invalid would require specific prompts to let the user know if they have filled in something wrong.



## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

- The system shall allow users to use the system immediately following registration.
- The system shall have an at least 99.9% uptime.
- The system shall have a search functionality that returns results within 5 seconds.

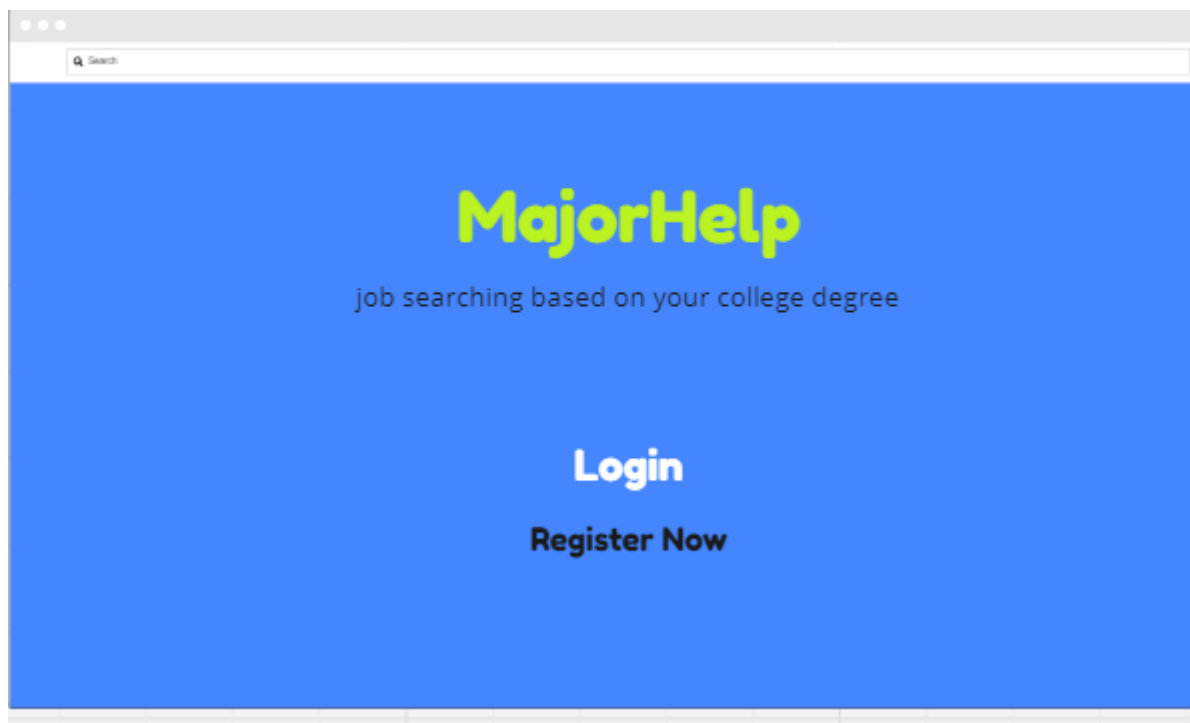
### 5.2 Safety Requirements

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>*

- The system shall be General Data Protection Regulation (GDPR) compliant.

### 5.3 Security Requirements

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication*



requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

- The system shall allow users to register so long as they have an email that is not previously registered.
- The system shall only allow a user to register if they have verified their email during the registration process.
- The system shall store user login information securely by using encryption.

## 5.4 Software Quality Attributes

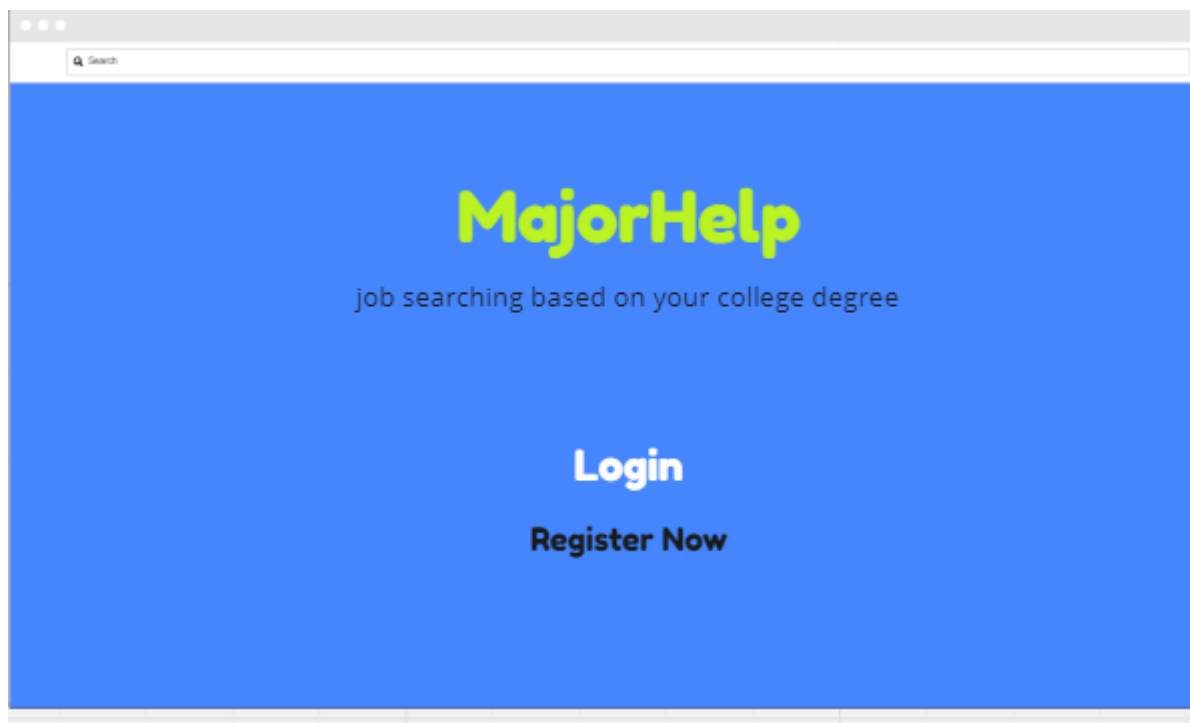
<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

- The system shall have a login and registration process that is intuitive so that nobody should have questions on how to login or register.
- The system shall be easily usable on mobile devices as well as desktop computers.
- The system shall allow students to filter their search based on a 5 star rating scale for companies.

## 5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

- The system shall allow users who are students to follow companies they are interested in.
- The system shall allow companies to create and “easy apply” form for students to fill out.



## 6. Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

### Appendix A: Glossary

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

### Appendix B: Analysis Models

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>*

### Appendix C: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*

