**Week 5 Final Project**

Braden Abbey

CST 499: Capstone for Computer Software Technology

Amr Elchouemi

10-31-2022

Table of Contents

Table of Contents ii

Revision History ii

1. Software Requirements Specification 3

1.1 Introduction 3

1.2 Overall Description 3

1.3 External Interface Requirements 5

1.4 System Features 5

1.5 Other Nonfunctional Requirements 10

2. Testing Levels and UML Models 11

2.1 Introduction 11

2.2 Integration Testing 12

2.3 Component Testing 12

2.4 System Testing 12

2.5 Acceptance Testing 12

2.6 UML Models 12

3. Test Management Strategy 16

3.1 Test Teams 16

3.2 Test Roles 16

3.3 Exit Criteria 17

3.4 Test Estimated Effort 17

3.5 Test and Risk 17

3.6 Incident Reporting 17

3.7 Defect Classification 18

3.8 Configuration Management 19

4. Test Tools 20

4.1 Introduction 20

4.2 Test Management and Control Tools 20

4.3 Test Specification Tools 20

4.4 Static Testing Tools 20

4.5 Dynamic Testing Tools 21

4.6 Non-Functional Testing Tools 21

5.0 Resources 21

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Braden Abbey | 10-03-2022 | Starting Document | 1.0. |
| Braden Abbey  Braden Abbey | 10-10-2022  10-31-2022 | Adding in Testing Levels and UML Models  Adding test management and test tools | 1.1  1.2 |

# Software Requirements Specification

## Introduction

### Purpose

### The purpose of this SRS document (specification document) is to outline the requirements that are to be used for the course enrollment system build. The new course enrollment system will allow students to be able to view and manage their class schedule. This newly created document will be version 1.0 and the scope of the SRS document will include the entire build of the course enrollment system. This SRS will also include basic features and non functional requirements.

### Document Conventions

This SRS document will be in written in 5 section those sections are introduction, overall description, external interface requirements, system features, and other non functional requirements. Sub header will also be present in each section to give more details on each section and explain in a throuough manner.

### Intended Audience and Reading Suggestions

In order to get an understanding of the requirements of the course enrollment system it is suggested that anyone reading should read the entire SRS. This SRS document is for use by UI/UX designers, project managers, software engineers, IT architects, testers, analysts, and other stakeholders that will be involved with the stand up of this project.

### Product Scope

The project goal is to allow students to manage their class schedule in a fast and easy manner. The website will allow students to create an account and register, log in, search for classes that are available, register for open courses, remove a course, get in on a waitlist for a course that has one, and or course view their schedule. The website will allow for an increase in enrollment, student retention, increase student experience, and will also reduce overhead costs from the manual process that is currently in place for registration. All of these benefits align with the business objective for the development of the course enrollment system.

## Overall Description

### Product Perspective

The Course enrollment system is a new build.

### Product Functions

With the course enrollment system a user should be able to register/create an account, log into their account, view their courses, search for any courses that are available, register for an available course, drop a course they no longer need, and be able to join a waiting list for a class if it is already full. If a class is already at max capacity then the website should ask the user if they want to be added to the waitlist for the course. The system should sent out a notification if there is an opening that becomes available for a course that a student is on the waitlist for.

### User Classes and Characteristics

Some of the user classes that will be present are students, system administrators, and employees at the school working with class registry. Student will need to be able to create and log into their new account, view course schedule, register/ search for available courses, and also drop a course as needed. The system administrators will need to be able to have full access to the system in order to help with any issues that may arise with the system. Some of these issues could be students having issues logging in and system type issues. The system administrators will also need to be able to look up student schedules by their assigned student id number. Employees at the school will need to have access to add classes, update classes, and delete classes. The primary user class for the system will by the student class and in addition to the classes outlined there will be another class for courses, this is used to give information about the course as well as the methods.

### Operating Environment

All students will not have the same operating system on their computers or on their cell phones. So the course enrollment system needs to be able to function on all major operating systems and web browsers. These include Internet Explorer, Edge, Chrome, Opera, IOS, Windows, Android, and Mac OS

### Design and Implementation Constraints

The Course enrollment system need to built in 8 months and have all bugs and defects worked out in a years time. All passwords for student accounts must use encryption and the course enrollment system must utilize a relational database. With feedback from students coming in all the time we need to make sure the system is scalable, maintainable, and supports modified content. We will build the Course Enrollment System to utilize ADA web accessibility guidelines. ADA is the Americans with Disabilities Act and helps to make sure people with disabilities have the same opportunities as everybody else (*What is the Americans with disabilities act (ADA)?* ADA National Network, 2022, August 31). Some tools that will be present in the Course Enrollment System will be text to speech, Color Assist, and magnifying glass.

### User Documentation

The Course Enrollment System will have a help section that students can access when they have a problem to either look at frequently asked questions or open a support ticket. The Course Enrollment system will also have live chat available for issues during main school hours, and there will be documentation available for viewing for help with tasks such as creating an account or registering for a course.

### Assumptions and Dependencies

Th Course Enrollment System has a budget of $500,000 USD and has a timeline of up and running in 8 months and defects worked out within a year. The project team will consist of 10 members and those are 1 project manager, 5 developers, 1 tester, 1 Analyst, 1 IT architect, and 1 database administrator.

## External Interface Requirements

### User Interfaces

All pages on the Course Enrollment system will use the same navigation bar at the top of the page, and will utilize icons from Flaticon for all icons on the system. The system will be built with no ads to not take attention from the students. Also each page will allow the student to return to a previous page and not lose any information from that page. The Course Enrollment System will also use colors that appealing to the eye and the home button will be visible on every page and will be easily distinguishable form the other icons. The help section should be nicely put together and guide students through basic problems and allow the use of documentation that is on the website.

### Hardware Interfaces

The Course Enrollment system will be able to be used on all major devices such as Cell phones, tablets, laptops, desktops, and any other system offered by Microsoft or Apple that students may use to access the Course Enrollment System. The Course Enrollment System website will make use of HTTP for making requests and TCP when moving resources to the client from the web server. HTTP stand for Hypertext Transfers Protocol and TCP stands for Transmission Control Protocol. HTPP is used when students access the website and TCP will be used when the session is establishing between the students client (computer/phone) and the website (Bhardwaj, 2021, January 19).

### Software Interfaces

The website should be Able to be used on Android, IOS, Windows, and Mac OS. The website build will also have an API that will talk to the relational database, this will be for retrieving and storing user data as well as course data. The sign on being used for the website should be the students school email address.

### Communications Interfaces

Students will receive an email when their support ticket is resolved or needs more information. When a student is using live chat and waiting in the queue they will get a notification to their phone or pc when it is their turn in line. Students will also receive email notification when they register for a class, drop a class, or when a class has a spot that has opened up. When emails get sent out all PII information should be secure and passwords for all users of the system must be encrypted.

## System Features

### New User Can Create an Account

Description and priority

* Priority: High
* Description: New user’s to the system should be able to create an account. You must have an account to view courses, view your schedule, apply for courses, and join a waitlist. This is a basic part of functionality for the website.
* Benefit: 10

Stimulus/Response Sequences

* When the user hits the landing page for the website they are prompted to log in or create an account
* User clicks on create account
* User enters in all information asked
* User clicks on create account
* User gets a message saying that they have created an account successfully, if an error occurs they will receive information on what is wrong.

Functional Requirements

REQ 1: Log in and Create account should be available on the home screen

REQ 2: When a user hits Create account they should be taken to the form to be filled out

REQ 3: The information in the form is validated and processed and then sent to the database when the user hits submit.

REQ 4: Website will let the user know if they account was created successfully or not

REQ 5: The user will get a success message when the account was made successfully and an error if not

### Returning User Can Login

Description and priority

* Priority: High
* Description: Any user coming back to the website should be able to log in as they need to log int o be able to look at schedule and move courses around. This is also basic functionalist of the website.
* Benefit: 10

Stimulus/Response Sequences

* User receives a prompt to log in or create an account once they hit the home page
* User clicks on the log in button
* User enters in username/password
* User clicks the log in button
* User gets taken into their account or receives an error is the information is incorrect.

Functional Requirements

REQ 1: Log in and Create account should be available on the home screen

REQ 2: When user clicks log in

REQ 3: The information in the form is validated and processed and then sent to the database when the user hits submit.

REQ 4: Website will let the user know if they have logged in successfully or if the information was incorrect.

REQ 5: The information stored in the database will be compared to the information the user inputs when logging in.

### User Can Search for Available Courses

Description and priority

* Priority: High
* Description: This is basic website functionality for the website. A student should be able to search for any available courses that they can take during each semester. In order to register for an open course a student must first be able to search for those courses.
* Benefit: 8

Stimulus/Response Sequences

* + User clicks on register for available courses which is available in their course schedule section
  + User selects their current semester from the drop down and clicks on submit
  + User is given a list of available courses to choose from

Functional Requirements

REQ 1: Website will have a register for courses link on the students current schedule

REQ 2: Website will give the student a drop down with courses available for the semester

REQ 3: When a user searches for a course the website should bring up available search results

REQ 4: Search results should be updated when a new course is searched.

### User Can View Course Schedule

Description and priority

* Priority: High
* Description: This is basic functionality of the website. The student should be able to view their schedule and look at their assigned courses. A user cannot change their schedule around without being able to view their current courses.
* Benefit: 9

Stimulus/Response Sequences

* + User clicks on view course schedule present in the top navigation bar
  + User is able to view their upcoming courses and full schedule

Functional Requirements

REQ 1: Navigation bar has a link to view the students course schedule

REQ 2: When the link for course schedule is clicked the student is redirected to a page that shows their current classes and their upcoming classes.

### User Can Register for a Course

Description and priority

Priority: High

Description: This is basic website functionality. A user/student should be able to register for any available courses that are not full. A user wont be able to manage their course schedule if they cannot add a course to that very schedule.

Benefit: 8

Stimulus/Response Sequences

* User clicks on the register for courses button that is present on the course schedule after if has been opened from the link on the top navigation bar
* User selects which semester they want to register a course for
* User is given a list with results for current and upcoming semesters
* User makes a selection next to every course they would like to take
* User hits add courses
* The user will be put into those courses if there is availability, they will receive a success message or a failure message
* If course is full the user has the option to be put on a waitlist
* The user will be brought to the course schedule page once they have successfully registered for courses and cleared all errors

Functional Requirements

REQ 1: Website should have a register for courses button that is present on the course schedule

REQ 2: Website should give the student a drop down list of all available semesters that they can add courses for

REQ 3: Website should return search results when the user makes a selection for which semester they want to add a course for.

REQ 4: Checkboxes should be present next to each class to make a selection

REQ 5: Website should have an add course button once selections are made

REQ 6: Website should show a success message if the student was able to register for the course, if not it should display an error message and explain why

REQ 7: The website should redirect the student back to their course schedule once they add their desires courses and fix all errors.

### User Can Drop a Course

Description and priority

* Priority: High
* Description: This is basic website functionality. The user should be able to drop a course they no longer want.
* Benefit: 9

Stimulus/Response Sequences

* + User clicks on view course schedule in the top navigation bar
  + User is able to view their course schedule
  + Users checks the check box for all classes they want to drop
  + User hits the submit button
  + User gets a success message if they have dropped the course successfully
  + Once the class is dropped the user is taken back to their course schedule

Functional Requirements

REQ 1: View course schedule is present on the top navigation bar

REQ 2: When the link is clicked the user is taken to the course schedule

REQ 3: Checkboxes should be present next to courses

REQ 4: Website should pull data from course table when the submit button is clicked

REQ 5: Website should give a confirmation message when the student successfully drops a course

REQ 6: Student/user should be taken back to the view course schedule landing page once they drop courses successfully

### User Receives Notification When Course is Available

Description and priority

* Priority: Medium
* Description: User should receive a notification to their device when a course is available that they are on the waiting list for
* Benefit: 8

Stimulus/Response Sequences

* + User logs into their account on the Course Enrollment System using their credentials
  + User receives a notification icon on the top navigation bar when a course on their waitlist is available
  + User is able to view notification and dismiss it

Functional Requirements

REQ 1: Website should allow the student/user to log in

REQ 2: Website should show the notification on the top navigation bar that a course they were on the waitlist for is available

REQ 3: User should be able to view notification and dismiss it

### User Can Logout

Description and priority

* Priority: High
* Description: This is basic website functionality. A user/student should be able to log out of the system. A user’s information is not full protected if they are not able to log out.
* Benefit: 9

Stimulus/Response Sequences

User clicks the log out button that is present on the top navigation bar

User is logged out successfully

Functional Requirements

REQ 1: Website should have a logout button on the navigation bar

REQ 2: Website should log the user out once the logout button has been clicked

REQ 3: Website should redirect the user to the home page once they log out successfully

## Other Nonfunctional Requirements

### Performance Requirements

When it comes to account creation a new user to the system should be able to set up an account in under a minute. When searching for classes results should be returned in less than 4 seconds. When adding or dropping a course a user should receive a notification email in less than 5 seconds. The system needs to have fast response time as if it is slow it can lead to users/students not returning to use it and if that happens it puts more work back on the administrators office.

### Safety Requirements

When the website is left idle for more than 15 minutes it will log out the current user. When making a new account there needs to be confirmation that the user is a student. When making accounts all passwords are to be encrypted and the student/user should be prompted to sign up for 2 factor authentication.

### Security Requirements

Passwords are all to be encrypted and students/users to sign up for 2 factor authentication. Logged in users will time out after 15 minutes of being idle and will be logged out of the system

### Software Quality Attributes

As previously stated the website and app should be available to everyone and needs to follow ADA guidelines. The website should have an up time as close to 99% as possible and be very responsive to those using it. The website needs to be scalable as there will always be more students coming and modifiable as there will always be room for enhancements to the system.

### Business Rules

A user needs to be logged in in order to be able to view their schedule and register/drop courses. Administrators will have the ability to troubleshoot issues for students such as resetting accounts and fixing any errors that occur. Administrators will also have access to search for student information by using the students assigned student ID as well as pull reports for anything they need such as class availability or students who added or dropped courses, etc.

# Testing Levels and UML Models

## Introduction

Software testing is something that is carried out through the entire lifecycle of a new software build. Testing helps to reduce costs, reduce development costs and time, and also improves maintainability and code quality. The customer will be getting a more robust and a piece of software free from most faults as they can be caught early in the lifecycle process by the user of testing. We are going to be using Integration testing, component testing, system testing, and lastly acceptance testing. Integration testing is where different units, modules, and components of a software application are tested together (Awati, 2022, January 13). Component testing is where components are tested independently with no other integrations. System testing is where we test a fully completed system with everything integrated and we look at end to end tests. Lastly acceptance testing is where we look at the system once done to make sure it is meeting the requirements, this is done to ensure we can deliver to the customer (*Acceptance testing*. Tutorials Point, n.d.).

## Integration Testing

Integration testing will need to be performed in order to make sure we can catch defects and errors between integrated components. Test cases will be made to test the integration of components, there are classes that will be shown below showing the sequence of events that will take place in order for a new student to register for a new course. Control flow and data flow will be a big part of integration testing.

## Component Testing

The new Course Enrollment System will have component testing that will occur for every different component. There will be various components as outlined below in the use case diagram that will have different functionalities. Some of these functionalities could be generating drop downs of classes, adding or deleting a course to a students course schedule, and displaying the students current classes on the course schedule. All classes and functions will be tested independently from one another as this is what component testing is all about, we test independently with no integrating at all. Component testing will let us improve maintainability, robustness, and effectiveness of the system for each functionality.

## System Testing

System Testing will take place once Component and Integration testing are completed, system testing is testing the system as a whole to make sure it meets requirements (Hamilton, 2022, August 27). We will be testing every aspect of the Course Enrollment system to what we gathered in the requirements phase such as availability, security, is the website responsive. System testing is testing the system from an end user’s perspective and is done in an environment that almost mimics production but is not using production data.

## Acceptance Testing

The last testing we have is Acceptance testing, The Course Enrollment System will be going through acceptance testing to make sure the system is built to requirements. The big functionalities will be gone over with the stakeholders to show the system is as they wanted it. Also in this step we will make sure we can generate reports, the live chat is working as expected, and courses are showing as they should for students when they are available. Acceptance testing will be carried out on multiple platforms and browsers to make sure it is working on all. Once we are finished with all testing we can ensure we are delivering a high quality product to the stakeholders.

## UML Models

Figure 1. Use Case Diagram

Diagram

Description automatically generated

Figure 2. Activity Diagram

Diagram, schematic

Description automatically generated

Figure 3. Class Diagram

Diagram, schematic

Description automatically generated

Figure 4. Sequence Diagram

Diagram

Description automatically generated with low confidence

# Test Management Strategy

## Test Teams

Test teams are needed for every project and the course enrollment system is no exception to this. Testing teams are needed in order to identify defects, reduce flaws in components and systems, and increase quality (Sharma, 2022, October 26). Test teams will include folks from the business side as well as IT. Specialists should be the ones doing system testing so we can test the system from different view points. It is also important to note that testers should not be testing any code that they wrote as they can turn a blind eye to it or not look at it from different perspectives.

## Test Roles

For testing the Course Enrollment System we will have multiple different Test Roles. We will have a test manager, 2 test designers, 2 test automators, and multiple testers. The test manager will be someone who has tons of experience in software testing and has a project management background. The automators will have experience in automation testing and will do the testing for automated test cases. The testers will be the QA Analysts that carry out test scripts and test cases. Along the way when they run into issues they will document defects.

## Exit Criteria

When it comes to asking the question if a test case is finished we need to look at exit criteria. Exit criteria is documented and has deadlines and conditions that must be met in order to be considered completed (*Entry and exit criteria in software testing*. ThinkSys Inc, 2022). An exit criteria will need to be in place for test cases for the course enrollment system. This is so we can know when to stop testing. The testing team will set the exit criteria when the project is first taking flight.

## Test Estimated Effort

When we start testing for the course enrollment system we will need to lay out a few items. These items are cost and time. How much money will this testing effort cost and how much time will be going into it from various resources and as a whole. Test effort will be handled by the project manager and need sto be handled during the planning phase. The reason behind this is to make sure there are resources and budget available. In order to make a solid estimation for testing efforts we will take a look at similar system builds to get a starting baseline. After we have a baseline we can do what is call work breakdown structure and break down the complex tasks into smaller ones to see better detail structure of the tasks that need done (*Estimation techniques in software testing [with examples]*. SaM Solutions, 2021, April 23).

## Test and Risk

When it comes to testing we don’t just go off and test anything we want in any order we want. When it comes to testing we will have some test cases that carry a higher risk than others when it comes to if a defect happens. The higher risk test cases should be tested first, for an example in the course enrollment system we would want to test the log in function before testing a top navigation bar. We want to make sure there are no downstream defects that will affect other systems. If we find these defects early we can save time and money by not having to troubleshoot them later on. Risk based prioritization is something that will come into paly for the Course Enrollment System. Some parts of the course enrollment system will be tested much more than others. Risk is calculate by loss or damage that could happen if that function were to not work properly. Risk management techniques will help us to mitigate risk and the biggest one is testing (Spillner, Linz, & Schaefer, 2014).  Testing allows us to see the system in action and we can grade the system on pass or fail when it comes to function. Risk based testing can be transformed into test cases and can target areas of risk as we see fit.

## Incident Reporting

The course enrollment system incident reporting will be for any issue or defect that is found in the system. Incident reporting is what we will use when we find issues during testing. Every defect will need to be documented in the below template to make sure we are gathering the correct information. Developers will be able to look at the defect information gathered and get a quick summary of what the issue is and what they can do to start fixing it. The information gathered during incident reporting will be testers name. test case number, environment, steps to recreate, priority of defect, short description, impact, and work around. Anyone on the project team can report an incident/defect, the work is being done by an engineer as code will need to be fixed. The incident/defect gets documented and closed once the fix has been put in place and the fix has been verified.

## Defect Classification

The Course Enrollment System will make use of defect classification to keep track of all the defects uncovered in the system. Defects have their own severity level as well as an impact level. The below chart gives a quick example of that and shows sev 1 is the highest with sev 5 being the lowest. Impact has a separate chart for letting us know how many end users have the issue occurring and goes as follows: Impact 1 is 20 or more user, impact 2 is 10-20, and impact 3 is anywhere from 0-10 users. Defect classification allows us to weigh defects for the Course Enrollment system so we can make sure the higher priority defects are being worked on first.

Table

Description automatically generated

(Spillner, Linz, & Schaefer, 2014).

Table

Description automatically generated

(Spillner, Linz, & Schaefer, 2014).

## Configuration Management

Configuration Management will make use of what is called configuration management, this will be used for tracking version history of the project. The reason we want configuration management is that multiple people can work in the same document without having to wait for the last person to save and close it, they can work at the same time. The same applies to environments like if an engineer is working at the same time as another they don’t need to fear of overwriting work. If we did not put configuration management in place we would see issues like unknown component versions, work being overwritten, and testing issues where components have been changed but the test team does know what test cases are for which version of the object (Spillner, Linz, & Schaefer, 2014). So for the Course Enrollment System we are going to make use of Version Management which is for filing and retrieving version numbers of a configuration item, like 2.1 versus 2.3 (Spillner, Linz, & Schaefer, 2014). We will also conduct quarterly configuration audits to make sure configuration management is being documented and are able to be identified.

# Test Tools

## Introduction

The Course Enrollment System is being made from scratch and doesn’t have anything out of the box from another company. We will need to make sure we are making use of what are called test tools in order to help guide us through our testing. Test tools make testing faster as we don’t need rely heavily on manual testing and we will see and increase in reliability and efficiency. Automation will play a big part here as it allows us to test thoroughly and much faster than the manual approach. The entire SDLC lifecycle of the Course Enrollment System will make use of testing tools for static, dynamic, non functional, test management, and test specification.

## Test Management and Control Tools

Test case management is important to the Course Enrollment system as it allows us to maintain our test cases. For this project we will make use of a tool called Jira, Jira allows us to connect all defects, requirements, and test cases to one another (Osuch, 2022, January 25 A good test management tool is needed to make sure that test cases are meeting the requirements that have been laid out. Now when it comes to control tools we will use these in automation for executing test scripts, the results will then be displayed in an easy to read format.

## Test Specification Tools

The specification phase will make use of testing tools such as test generators. Test generators will be used in database code, specification, and interface. Test generators for database are used to for processing schemas and producing test databases from the schemas used (Spillner, Linz, & Schaefer, 2014). Test data is produced by using existing code with code based generators, code must be present in order to be used for this. Finally specification based test generators are used to pull test data from a desired specification that is used. Test scripts get generated then go down to the test execution tool (Spillner, Linz, & Schaefer, 2014). Zephyr is an excellent choice for a test specification tool as its easy to use and flexible. It will allow us to integrate with Jira and will help with improving visibility and collaboration on projects (*Top 20 best test management tools (new 2022 rankings)*. Software Testing Help, 2022, September 24).

## Static Testing Tools

Static testing is great to use as we don’t need to execute the program in order to do it. The Course Enrollment System will use static testing on its code which will help us in finding defects early. Review tools are used in static testing and a great one to look at is called Raxis. Raxis is a static testing tool and is used for finding time wasters known as false findings (*Top 40 static code analysis tools (best source code analysis tools)*. Software Testing Help, 2022, September 24). Static testing should be used to look at complex areas of code as they carry more risk and should prioritize them in the testing process.

## Dynamic Testing Tools

For dynamic testing the Course Enrollment system will make use of simulators, test robots, debuggers, and dynamic analyzers. When using these tools we will need top make up test data for them to use as an input and they will watch it go through and give us the output. We will run our code through an IDE which will do debugging as it takes each line of code and analyzes it, IDES can be started and stopped as we need. Simulators will allow us to test application flows with no need to try and mimic hardware set ups. Simulators will save us a lot of money in hardware costs. Test robots can benefit us by looking at manual inputs and keyboard strokes to help pinpoint problems and test. The test robot will store this information in a script and we can then take that script and go through it to see the actions done with automation. Dynamic analyzing tools are last here and they will help to analyze code while it is running. An example of this would be buying a dirt bike, dynamic analysis would be taking a test drive and see how it handles in corners, how the clutch feels, engine oil fill level and even compression. It is the backend components that can only be seen while its running.

## Non-Functional Testing Tools

For the Course Enrollment System we are going to need to make use of non-functional testing tools. These tools are used to verify non functional requirements such as usability and performance. The reason is to make sure the system is working as per the requirements written out. These will include any requirements not covered by functional testing and requirements gathering sessions. These are the behavior of the system when under load and will be documented as user stories and technical stories as acceptance criteria and as well as in artifacts. Loadster would be a good program that would be good for looking at load and tress testing. It will allow us to take load scripts and keep track of them in the browser and then step through the website and mimic a real end user (*Load & stress testing for high-performance websites*. Loadster, n.d.).

# 5. Resources

*Acceptance testing*. Tutorials Point. (n.d.). Retrieved October 7, 2022, from https://www.tutorialspoint.com/software\_testing\_dictionary/acceptance\_testing.htm

Awati, R. (2022, January 13). *What is integration testing (I&T)?* SearchSoftwareQuality. Retrieved October 7, 2022, from https://www.techtarget.com/searchsoftwarequality/definition/integration-testing

Bhardwaj, R. (2021, January 19). *HTTP VS TCP : Detailed comparison* . Network Interview. Retrieved August 28, 2022, from <https://networkinterview.com/http-vs-tcp-know-the-difference/#:~:text=HTTP%20is%20a%20Hypertext%20Transfer%20Protocol%2C%20whereas%20TCP%20full%20form,TCP%20uses%20the%20TCP%2DAO>.

*Entry and exit criteria in software testing*. ThinkSys Inc. (2022, May 5). Retrieved September 11, 2022, from https://www.thinksys.com/qa-testing/entry-exit-criteria/#:~:text=What%20is%20An%20Exit%20Criteria,end%20of%20software%20testing%20process.

*Estimation techniques in software testing [with examples]*. SaM Solutions. (2021, April 23). Retrieved September 16, 2022, from https://www.sam-solutions.com/blog/what-are-the-estimation-techniques-in-software-testing/

Hamilton, T. (2022, August 27). *What is system testing? types with example*. Guru99. Retrieved October 7, 2022, from https://www.guru99.com/system-testing.html

*Load & stress testing for high-performance websites*. Loadster. (n.d.). Retrieved September 25, 2022, from https://loadster.app/

Osuch, B. (2022, January 25). *Benefits from using jira for software tests*. Deviniti. Retrieved September 24, 2022, from https://deviniti.com/blog/application-lifecycle-management/jira-testing-tool/#:~:text=Requirements%20and%20Test%20Management%20for%20Jira%20(RTM)%20is%20a%20fully,and%20reporting%2C%20right%20inside%20Jira.

Sharma, L. (2022, October 26). *Why is testing necessary and important?: ISTQB*. TOOLSQA. Retrieved October 26, 2022, from https://www.toolsqa.com/software-testing/istqb/why-is-testing-necessary/

Spillner, A., Linz, T., & Schaefer, H. (2014). [Software testing foundations: A study guide for the certified tester exam (4th ed.)](https://ashford.instructure.com/courses/106377/modules/items/5401778).

*Top 20 best test management tools (new 2022 rankings)*. Software Testing Help. (2022, September 24). Retrieved September 25, 2022, from https://www.softwaretestinghelp.com/15-best-test-management-tools-for-software-testers/

*Top 40 static code analysis tools (best source code analysis tools)*. Software Testing Help. (2022, September 24). Retrieved September 25, 2022, from https://www.softwaretestinghelp.com/tools/top-40-static-code-analysis-tools/

*What is the Americans with disabilities act (ADA)?* ADA National Network. (2022, August 31). Retrieved August 28, 2022, from https://adata.org/learn-about-ada#:~:text=The%20purpose%20of%20the%20law,origin%2C%20age%2C%20and%20religion.