

# Software Requirements Specification

Version 1.0

<<Annotated Version>>

May 15, 2021

Uedmy Support App

Anmar Almerie

Nouar Sabouh

Submitted in partial fulfillment Of  
the requirements of  
CS 310 Software Engineering

<<Any comments inside double brackets such as these are *not* part of this SRS but are comments upon this SRS example to help the reader understand the point being made.

Refer to the SRS Template for details on the purpose and rules for each section of this document.

This work is based upon the submissions of the Spring 2004 CS 310. The students who submitted these team projects were Thomas Clay, Dustin Denney, Erjon Dervishaj, Tiffanie Dew, Blake Guice, Jonathan Medders, Marla Medders, Tammie Odom, Amro Shorbatli, Joseph Smith, Jay Snellen, Chase Tinney, and Stefanie Watts. >>

## Table of Contents

Table of Contents .....	i
List of Figures .....	ii
<b>1.0. Introduction .....</b>	<b>4</b>
1.1. Purpose .....	4
1.2. Scope of Project .....	4
1.3. Glossary .....	5
1.4. References .....	5
1.5. Overview of Document .....	5
<b>2.0. Overall Description .....</b>	<b>7</b>
2.1 System Environment .....	7
2.2 Functional Requirements Specification .....	7
User Use Cases .....	9
2.2.1 Use case: Update Author .....	10
2.2.2 Use case: Update Reviewer .....	11
2.2.3 Use case: Update Article .....	12
2.2.4 Use case: Receive Article .....	10
2.3 User Characteristics .....	13
Non-Functional Requirements .....	13
<b>Requirements Specification .....</b>	<b>14</b>
3.1 External Interface Requirements .....	14
3.2 Functional Requirements .....	14
3.2.1 Search courses .....	14
3.2.2 Choose category .....	15
3.2.3 Browse the latest courses .....	15
3.2.4 Use a filters .....	16
3.3 Detailed Non-Functional Requirements .....	17
3.3.1 Logical Structure of the Data .....	17
3.3.2 Security .....	17

## List of Figures

Figure 1-System Environment .....	7
Figure 2-User use cases .....	8

## 1.0. Introduction

### ***1.1. Purpose***

The purpose of this document is to present a detailed description of the UdemY Support App. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

### ***1.2. Scope of Project***

This software system will be a App for the latest published courses for all students and developers around the world. This App will be designed to Alert the students about the latest courses on UdemY E-Lerning Platform and for the developers about the new courses with long list of categories like marketing, business, and more.

The users will have to deal with sample UI and filters that deal with rating, duration and prices of the courses.

### ***1.3. Glossary***

<b>Term</b>	<b>Definition</b>
Available Course	The Course that is tracked by the system; it is a sample content that is planned to be posted to the App.
platform	The platform that have the specific courses the will be provided.
Database	Collection of all the information monitored by this system.
User	A person that downloaded the App and start to use it.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.
User	Reviewer or Author.

### ***1.4. References***

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

### ***1.5. Overview of Document***

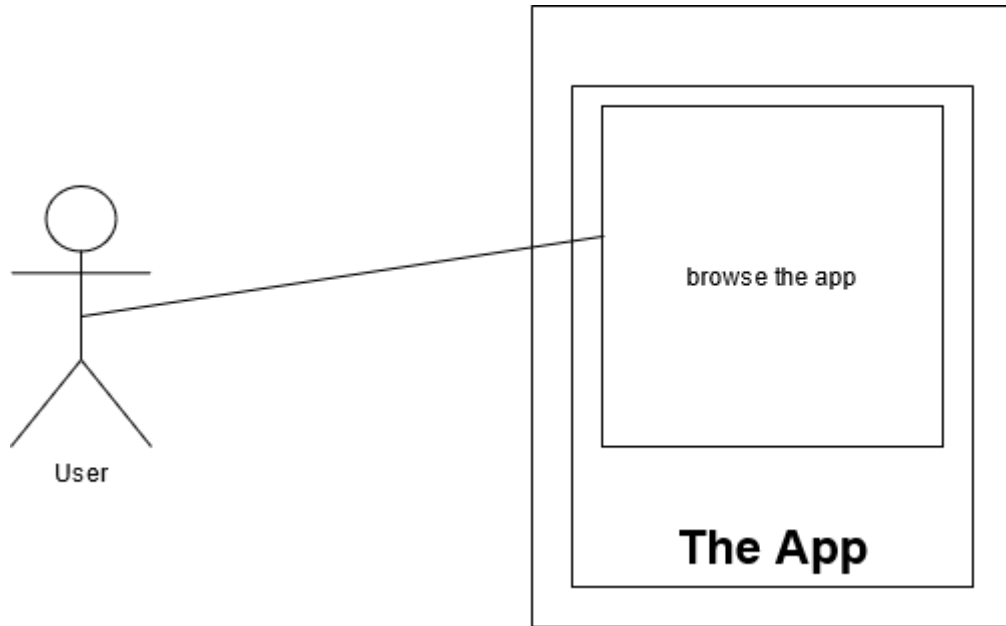
The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

## 2.0. Overall Description

### 2.1 *System Environment*



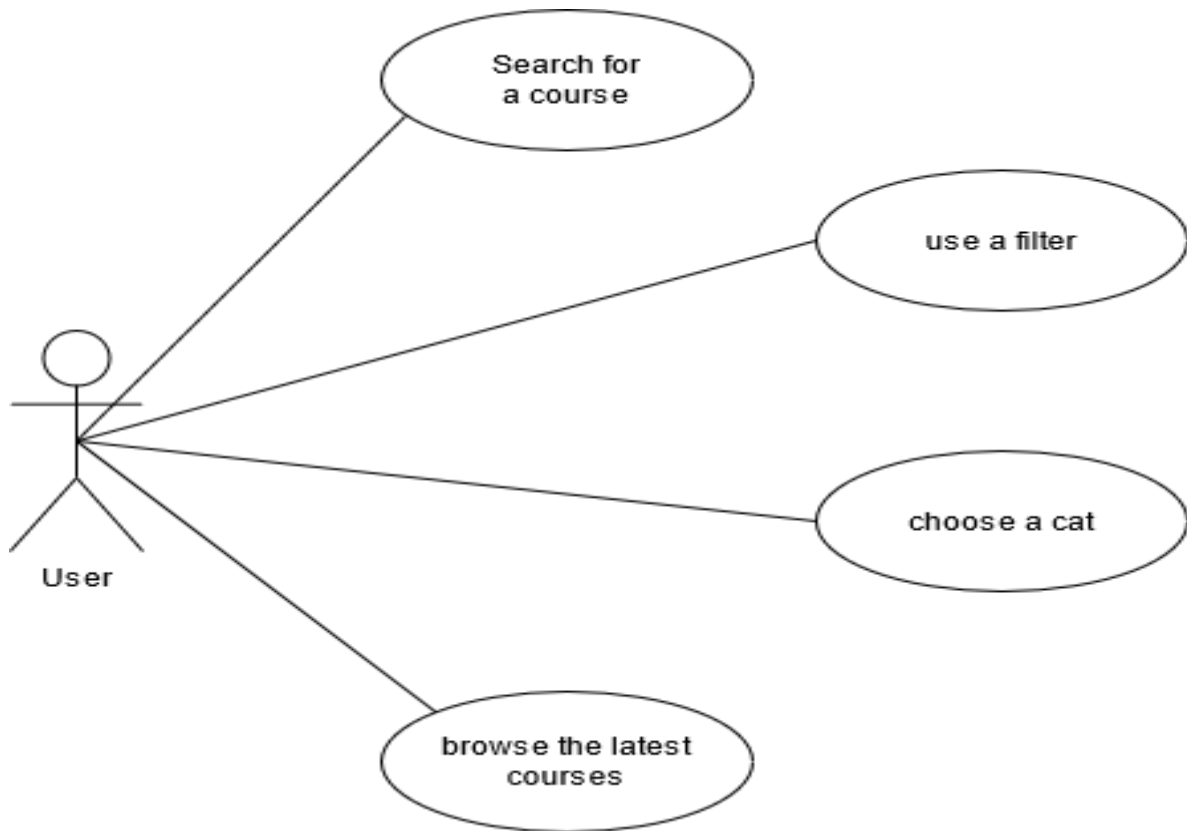
The App has one active actor which is the user of it, the user will browse the app and search for The specify course that he want, the app is connected with API that get the information from the data base.

### 2.2 *Functional Requirements Specification*

This section outlines the use cases for The user in this system.

User use cases:

The Use has the following sets of use cases:

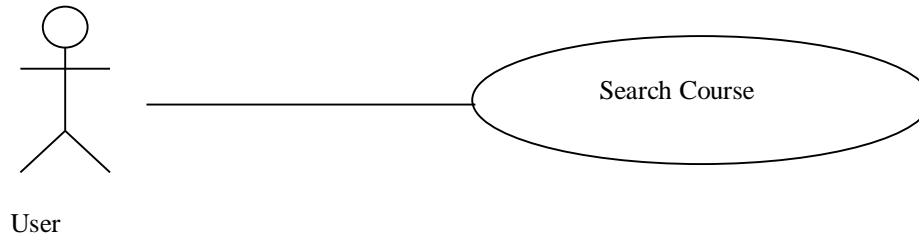




## Section 2.2.1

Use case: **Search Course**

**Diagram:**



### **Brief Description**

The user will tap on the search field at the bar and use it to search

### **Initial Step-By-Step Description**

Before this use case can be initiated, the User needs to access the app.

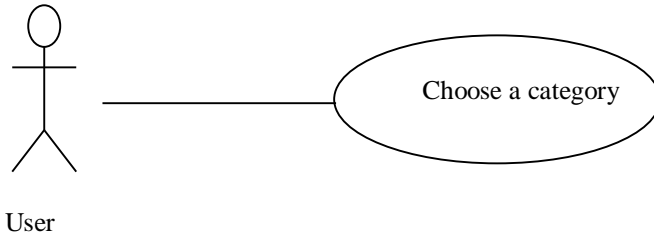
1. After the user get into the App he will taps on the search field at the top bar.
2. Then he will press on the search icon.
3. The system displays the choices to the User.
4. The User selects the course desired.
5. The User chooses to enroll the course that he want
6. The system will show the course in sample page.

**Xref:** Section 3.2.1, Search Course

## Section 2.2.2

Use case: **Choose a cat**

**Diagram:**



### **Brief Description**

The User will open the category page and choose one of the categories to see the latest courses in it.

### **Initial Step-By-Step Description**

Before this use case can be initiated, the User needs to access the app.

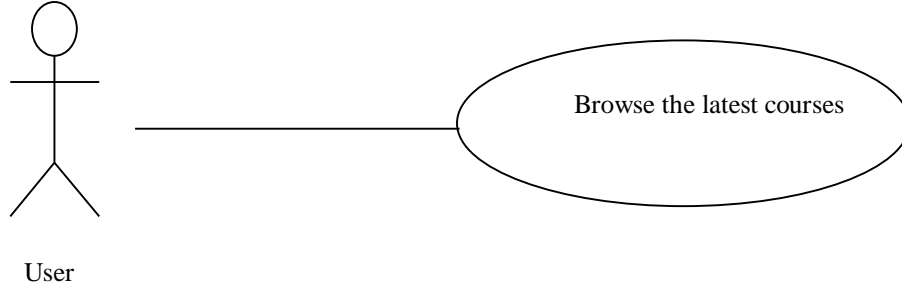
1. The User chooses to taps on the categories button on the top bar.
2. The System will show the list of categories that he has.
3. The User will choose one of the categories to check the latest courses in it
4. The System are going to show him the results.
5. The User chooses to enroll the course that he want
6. The system will show the course in sample page.

**Xref:** Section 3.2.2, choose a category.

### Section 2.2.3

Use case: Browse the latest courses

**Diagram:**



#### **Brief Description**

The user will scroll down and up the home page to check the latest courses.

#### **Initial Step-By-Step Description**

Before this use case can be initiated, the User needs to access the app.

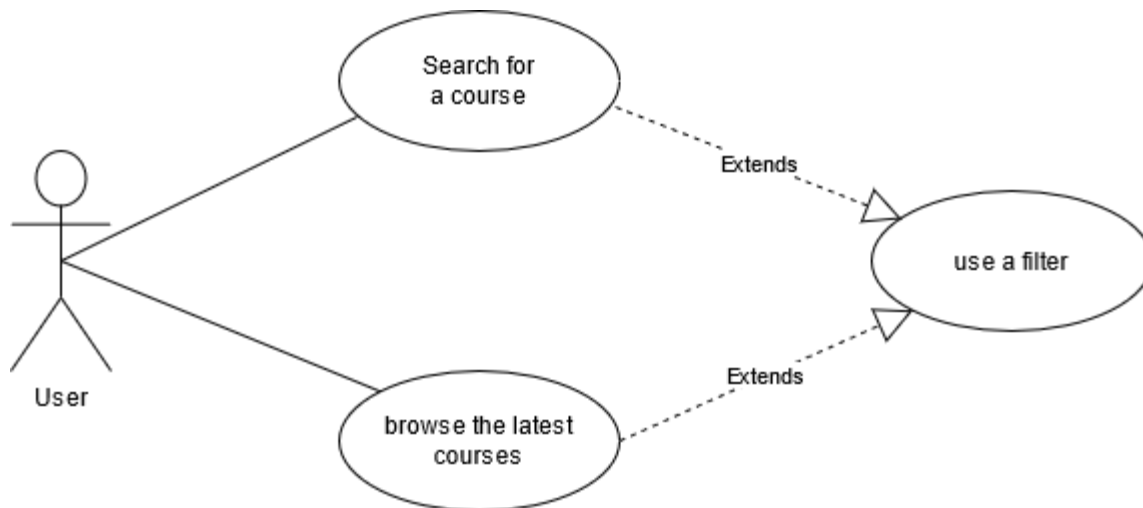
1. Directly after the user open the app the system will show him the latest added courses on the home screen.
2. The use will scroll down and up to check the available courses.
3. The User chooses to enroll the course that he want
4. The system will show the course in sample page

**Xref:** Section 3.2.2, Browse the latest courses

## Section 2.2.4

Use case: Use a Filter

**Diagram:**



### **Brief Description**

The user will use the filters to get a specify courses

### **Initial Step-By-Step Description**

Before this use case can be initiated, the User needs to access the app.

1. After the user access the app he got a several options search for a course, check the latest courses or choose one of the categories and get the latest added courses in it.
2. In all previous cases the check box of the filters on the top bar are going to show up
3. After the user mark the wanted filter the screen will refresh.
4. The system will show the results for the user.
5. The user will click on the wanted course

**Xref:** Section 3.2.3, use filters

### **2.3     *User Characteristics***

The User is expected to be Internet literate and be able to use the app.

The app home screen will guarantee the best user experience

### **2.4     *Non-Functional Requirements***

Instead of the native programming language the app will be coded with cross-platform programming tool (flutter).

That's how the app will be able to download through both google play and apple store.

The app are going to be hosted on modern host provider that provide SSL Security license.

### 3.0. Requirements Specification

#### 3.1 *External Interface Requirements*

The only link to an external system is the link to the App Database on the host

There is no log in data there is only cookies to verify the user data.

The data base is connected to the app through the API (node.js).

#### 3.2 *Functional Requirements*

##### 3.2.1 Search for a course

<b>Use Case Name</b>	Search for a course
<b>XRef</b>	Section 2.2.1, Search course
<b>Trigger</b>	The user assesses the app and taps on the search icon
<b>Precondition</b>	The app is displayed with grids for searching

<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The user will search by keyword.</li> <li>2. The app will show list of the latest added courses that got the same keyword that the user type in the search field.</li> <li>3. The user selects a course.</li> <li>4. The user enrolls in the course.</li> </ol>
<b>Post condition</b>	The selected course will redirect the user to the course url.
<b>Exception Paths</b>	The user may abandon the search at any time.

### 3.2.2 choose category.

<b>Use Case Name</b>	Choose category
<b>XRef</b>	Section 2.2.2, choose category
<b>Trigger</b>	The user selects the categories icon
<b>Precondition</b>	The user will tap on top bar section.
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The user will tap on the categories icon on the top bar.</li> <li>2. A list of categories will show up.</li> <li>3. The user selects on of the categories.</li> <li>4. The system will show a list of the latest courses that had been added in the specify category.</li> <li>5. The user selects a course.</li> <li>6. The user enrolls in the course.</li> </ol>
<b>Post condition</b>	The system will show a list of categories.
<b>Exception Paths</b>	The user may be abandoned at any time.

### 3.2.3 Browse the latest courses

<b>Use Case Name</b>	Browse the latest courses
<b>XRef</b>	Section 2.2.3, browse the latest courses
<b>Trigger</b>	The user access the app
<b>Precondition</b>	The user has accessed the app main screen.

<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The user access the app and get in the home screen.</li> <li>2. The system are going to show the latest courses that been added to the data base.</li> <li>3. The user selects a course.</li> <li>4. The user enrolls in the course.</li> </ol>
<b>Post condition</b>	The system will show a list of courses.
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>1. The User may abandon the operation at any time.</li> <li>2. The User may tap on the search icon</li> <li>3. The user may tap on the categories icon</li> </ol>

### 3.2.4 Use the filters

<b>Use Case Name</b>	Use the filters
<b>XRef</b>	Section 2.2.4, use the filters
<b>Trigger</b>	The User choose to use filters checkboxes to get a specify courses
<b>Precondition</b>	The User has accessed the app and taps on search bar options
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The User will access the app and taps on top bar options.</li> <li>2. He will choose the wanted filters to use</li> <li>3. The system will refresh the screen and show him the result.</li> <li>4. The user selects a course.</li> <li>5. The user enrolls in the course.</li> </ol>
<b>Post condition</b>	There must be courses in the data base to show
<b>Exception Paths</b>	<ol style="list-style-type: none"> <li>1. The user can access the categories screen then use the filters</li> </ol>
<b>Other</b>	There is no filters support with the search use case.



### 3.3 Detailed Non-Functional Requirements

The Logical Structure of the data to be stored in the App database on the server is as follows:

#### Published Article Entity

Data Item	Type	Description	Comment
Title	Text	Title of the course	
Author	Text	Name of one Author	May be several
Content	Text	Body of course	
Price	Number	Price of the course	Free Or paid
Category	Text	Area of content	May be several

#### 3.3.2 Security

The Host on which the App will have its own security to prevent unauthorized *write/delete* access. There is no restriction on browse access.

The provided host will gave a private security access like SSL and DHMRC



