SpartaDocs

Created by MASSware

CS 160 Section 2 Project

Team: Frank Daniels, Malik Khalil, Arjun Nayak, Sajay Shah, Shweta Sugnani

Prof. Ahmad Yazdankhah

Department of Computer Science

San Jose State University

Fall 2016

## Table of Contents

### Introduction 3

#### Purpose of System 3

#### Scope of the System 3

#### Acronyms and Abbreviations 3

#### References 3

### Requirements 4

#### Functional Requirements 4

#### Nonfunctional Requirements 4

### Use Cases 5

### Mockups 10

Dataflow 13

## Table of Figures

### Use Case 1 5

### Use Case 2 6

### Use Case 3 7

### Use Case 4 8

### Mockup 1 10

### Mockup 2 10

### Mockup 3 11

### Mockup 4 11

### Mockup 5 12

### Dataflow 13

# Introduction

## Purpose of the System

The purpose of SpartaDocs is to make the current graduation application and semesterly hold release processes paperless and minimize interaction with faculty. The current system requires one to fill out a form and access previous versions of documents. SpartaDocs will save time and money for both students and faculty by streamlining this process.

## Scope of the System

This system should be able to a large amount of users and administrators access different parts of the application. The system should have adequate resources to handle large amounts of requests from students and faculty for different internal resources. The system should be able to handle a version control system for multiple documents per student.

## Acronyms and Abbreviations

N/A

## References

N/A

# Requirements

##### Functional Requirements:

###### The User can complete forms through a user interface.

###### The User can request a previous version of their document to be retrieved.

##### Nonfunctional Requirements:

###### The application should be available to faculty and students at all times on any machine they choose.

###### All operations and changes made on the application should be viewable immediately, or at least within a reasonable amount of time.

Documents should be made available only to the people involved, and there should be zero loss in data.

###### 

# Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **First Use Case Name** | | Sign Up New User | |
| Goal | | | |
| This use case will describe how a user will sign up for SpartaDocs | | | |
| Participating Actors | | | |
| SJSU student | | | |
| Nonfunctional Requirements | | | |
| For example:   1. User should use a modern web browser. 2. User should have access to their individual SJSU email ID. 3. System should respond in a timely matter. (< 3 seconds) | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User clicks “Sign Up” button | | | |
| Primary Preconditions | | | |
| User must have access to email inbox | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User enters first name, last name, student ID, SJSU email address, and new password into sign up form. | | None. |
| 2 | User clicks “Sign Up” button. | | 1. System checks that no data fields are empty.  2. System checks if email address is valid  3. System creates a verification link.  4. System emails the verification link to the email address the user entered.  5. System redirects to log in page |
| Primary Postconditions | | | |
| User database will update with the input data. | | | |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| User Account Creation Fails | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | N/A | | notify user that account creation failed, ask user to try again. |
| Alternate Postconditions | | | |
| N/A | | | |

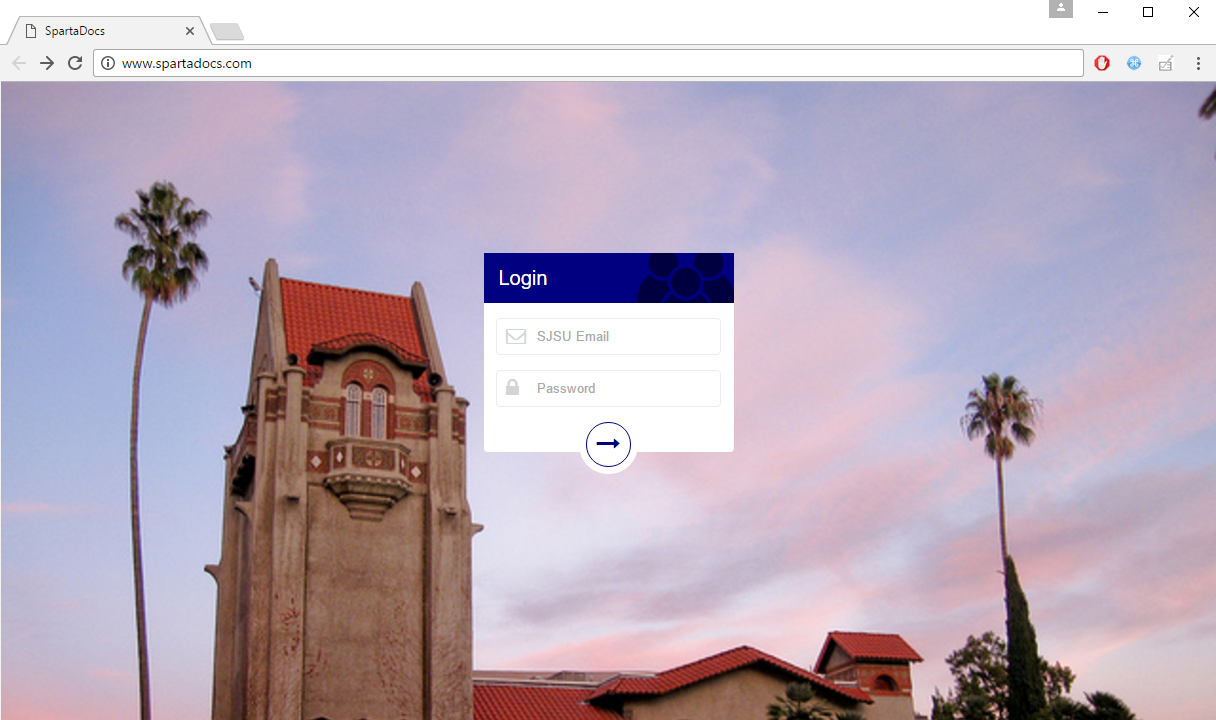
|  |  |  |  |
| --- | --- | --- | --- |
| **Second Use Case Name** | | Log In User | |
| Goal | | | |
| This use case describes how a user will log in into the system. | | | |
| Participating Actors | | | |
| SJSU student who has verified email with SpartaDocs | | | |
| Nonfunctional Requirements | | | |
| For example:   1. User should use modern web browser. 2. System should respond within a timely matter. ( < 10 seconds) | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User clicks on Log In button | | | |
| Primary Preconditions | | | |
| User should have already signed up for SpartaDocs and verified their email | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User enters their email, password in log in page. | | None |
| 2 | User clicks log in. | | 1. System validates entry.  2. If invalid combination, system will alert user to try again.  3. If valid combination then system will load welcome screen. |
| Primary Postconditions | | | |
| N/A | | | |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| N/A | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | N/A | | N/A |
| Alternate Postconditions | | | |
| N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Third Use Case Name** | | User starts hold-release form | |
| Goal | | | |
| This use case describes a user filling out a hold-release form | | | |
| Participating Actors | | | |
| Verified and authenticated SpartaDocs user | | | |
| Nonfunctional Requirements | | | |
| 1. System should respond in a timely matter. ( < 10 seconds) | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User clicks “Begin New Hold-Release Form” button on control panel. | | | |
| Primary Preconditions | | | |
| User is logged into the system | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User clicks Hold-Release form in control panel | | If Hold-Release form for the current semester has already began, all information will be automatically filled. The user will pick up from where they started. If not, an empty Hold-Release form will appear. |
| 2 | User fills out Hold-Release form accordingly. | | System updates the hold release object model |
| 3. | If User clicks “Submit” button | | System attempts to save form to database. Will notify user if it was successful or not and then redirect to dashboard. |
| Primary Postconditions | | | |
| Hold-Release form database for the user will save the appropriate data | | | |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| Form Failed To Save | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | Redirect user to dashboard | | notify user that form failed to save |
| Alternate Postconditions | | | |
| N/A | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fourth Use Case Name** | | User starts graduation application | |
| Goal | | | |
| This use case describes a user filling out a graduation application | | | |
| Participating Actors | | | |
| Verified and authenticated SpartaDocs user | | | |
| Nonfunctional Requirements | | | |
| 1. System should respond in a timely matter. ( < 10 seconds) | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User clicks “Begin New Graduation application” button on control panel. | | | |
| Primary Preconditions | | | |
| User is logged into the system | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User clicks Graduation application in control panel | | Empty Grad form appears |
| 2 | User fills out Graduation application accordingly. | | Grad Form Data Model will update accordingly |
| 3 | User clicks “Submit” button | | System attempts to save form to database. Will notify user if it was successful or not and then redirect to dashboard. |
| Primary Postconditions | | | |
| Graduation application database for the user will save the appropriate data | | | |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| Form Failed To Save | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | Redirect user to dashboard | | notify user that form failed to save |
| Alternate Postconditions | | | |
| N/A | | | |

# Mockups

Mockup 1: Login Screen



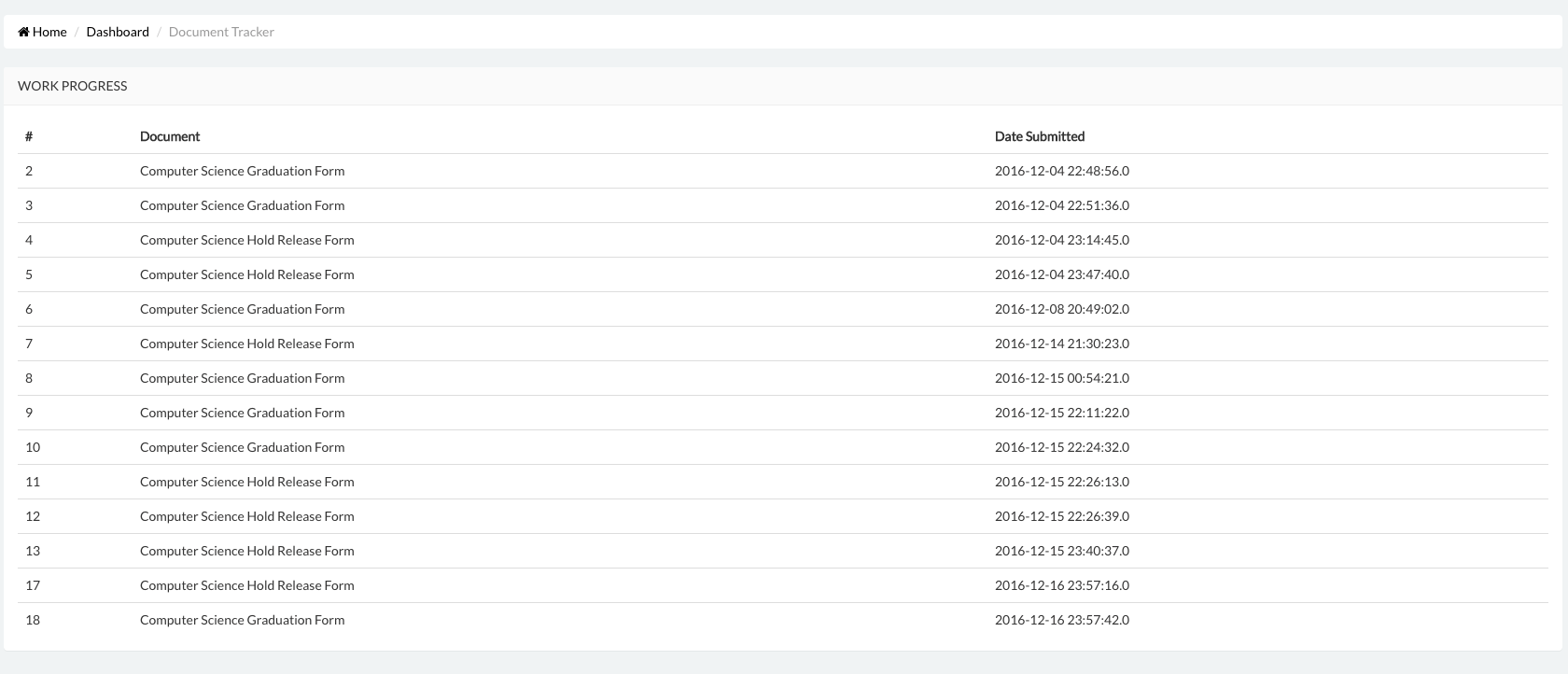
Navigation: Login leads to dashboard

Mockup 2: Dashboard

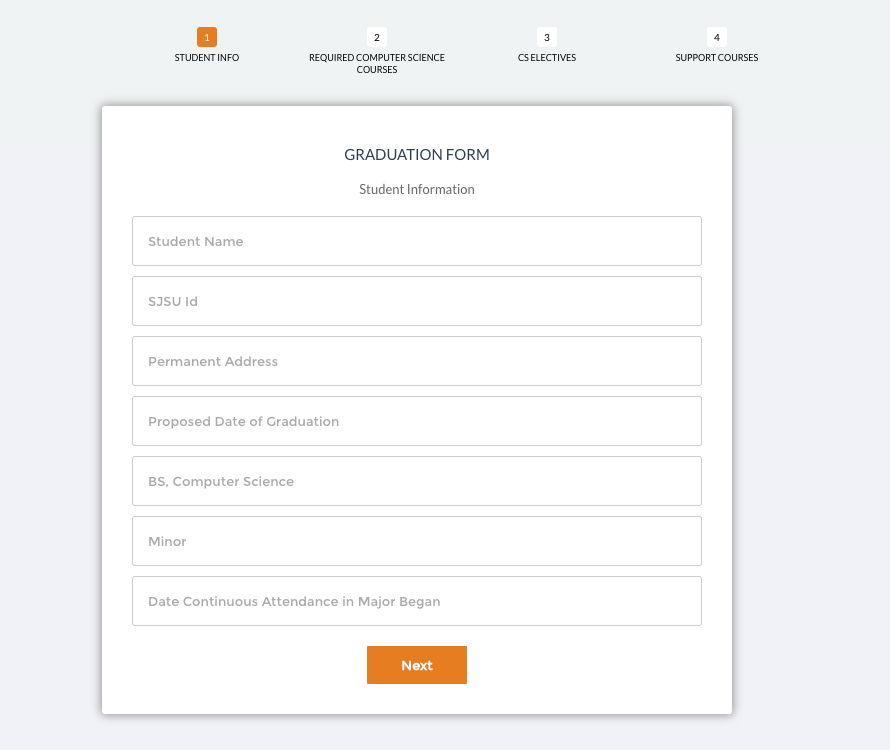
## 

Navigation: Document Tracker > Document Tracker Screen, Hold Release, Graduation Form > Appropriate forms

Mockup 3: Document Tracker



Mockup 4: Sample Graduation Form



# Dataflow