**Project Report #3**

**Team:** Codework Digital

**URL:** **thetalkmadesimple.com**

**Team Members:** Cat Schnelle, Chanse Stephens, Kevin Sims, Elijah Pemberton

**Group #:** 4

**Course:** CSCI-441-VA-Software Engineering

**Individual Contributions Breakdown**

**Report #3**

Chanse Stephens

* Updated Functional Requirements Specification Actors
* Updated Use Case descriptions
* Updated Key Accomplishments
* Updated User Interface Design and Implementation
* Updated Design of Tests
* Updated Summary of Changes
* Customer Statement of Requirements
* Interactive Diagrams
* Class Diagram
* Domain Analysis
* Traceability Matrix
* Architectural Styles
* Persistent Data Storage
* Network Protocol
* Global Control Flow
* Hardware Requirements
* User Interface Design and Implementation
* Project Coordination and Progress Report
* Individual Contributions Breakdown
* References

Cat Schnelle

* Report formatting
* Updated Summary of Changes
* Updated Current Status
* Updated User Interface Design and Implementation
* Updated Class Diagram and Interface Specification
* Updated Domain Analysis
* Table of Contents
* Project Management (Gantt chart + Trello board)
* Use Case Diagram
* Interactive Diagrams
* Functional Requirements Specification
* Domain Analysis
* Traceability Matrix
* System Operation Contracts
* UML Package Diagram
* Identifying Subsystems
* Design of tests
* Functionality Tests
* Useability Tests
* Interface Tests
* Compatibility Tests
* Integration Testing
* Individual Contributions Breakdown
* Plan of work / roadmap

Elijah Pemberton

* Future Status
* Editing, grammar correction
* Design of Tests
* Integration Testing

Kevin Sims

* Editing, grammar correction
* Domain Analysis
* Additions to data category

Table of Contents

Cover page and Individual Contributions Breakdown

Table of Contents

Summary of Changes

Customer Statement of Requirements

Glossary of Terms

Functional Requirements Specification

* Stakeholders / Actors & Goals
* Use Cases
* Use Case Diagram

Use Case Points

Domain Analysis

* Domain Model
* Concept Definitions
* Association Definitions
* Attribute Definitions
* Traceability Matrix
* System Operation Contracts

Interaction Diagrams

Class Diagram and Interface Specification

* Class Diagram
* Data Types and Operation Signatures
* Traceability Matrix

System Architecture and System Design

* Architectural Styles
* Identifying Subsystems
* Mapping Subsystems to Hardware
* Persistent Data Storage
* Network Protocol
* Global Control Flow
* Hardware Requirements
* Project Management

User Interface Design and Implementation

Design of Tests

History of Work

* Merging the Contributions from Individual Team Members
* Project Coordination and Progress Report

References

**Summary of Changes**

* Updated Functional Requirements Specification Actors.
* Updated Use Case descriptions.
* Updated Interface Design and Implementation
* Updated Design of Tests
* Updated Current Status
* Updated Class Diagram and Interface Specification
* Updated Domain Analysis
* Updated Key Accomplishments

**Customer Statement of Requirements**

I am a therapist with a private practice in Austin, Texas. Over my years of experience I’ve gained a multitude of knowledge while working with families. I’ve worked with families from all kinds of backgrounds including those involved with Child Protective Services. A lot of what I dealt with was helping parents or caretakers talk with their children about important topics in life. With all my experiences, I feel that I have a duty to share this expertise with the rest of the world. The best way I feel to do this is through an online resource. My plan is to have a website built that provides online courses for parents to learn how to talk about these things with their children. It is very important that parents and their children have access to resources and valuable knowledge at all times of the day.

Inspiration for the website is based on other successful platforms that have similar key functions. For example, a website like Masterclass which is an easy to use learning platform. Masterclass uses a simple interface to allow users to interact with the video content. Each video has a brief label, informing the user of what they can expect from the course. It does not overburden them with too much content information regarding the topic up front. Instead it uses simplicity to invite the user to access the content, then gradually introduces the information to the user. The way this website functions is inspiration for the functionality of the site due to its modernity and simplicity. With respect to layout and user interface, it is extremely accessible to any user based on the way video media is displayed on the website. Since this website is designed to be used for educational courses, this will be a helpful point of inspiration to study to ensure that the Codework Digital project attains the same level of ease of use. Another inspiration for the website are similar services that offer similar therapeutic information to users. For example, Better Help is a therapy application that pairs clients with licensed therapists online through chat or phone. While our website will not have the same functionality or services, when it comes to the need for privacy and making sure the agenda for providing this content is clear to the user and is easy to understand, this site can serve as inspiration for our services that are much simpler to offer with less liability. This helps us to plan ahead for any potential liabilities with privacy and security.

Functionality is extremely important. For the website to have the best possibility for success it must function smoothly, be easy to navigate, and simple to use. It is important that the site does what it needs to do in the simplest methods possible in order to meet the criteria for delivering us the content it promises. It should be simple and elegant in the way the content is presented and not going to burden me with too much information. The website should maintain my privacy by requiring only an email address to create an account. The account creation is used to access paid content and possibly other future features. I should only be contacted by the site through their email for site related updates for new content. The site should be easy to navigate for anyone accessing paid or free content and should be design consistent in terms of appearance and branding. The site should function on any web based platform or device and should be accessible with the exact same content whether it be videos or photo media regardless of what device they use. The site should also have tracking functionality that way the site owner can best cater their content to the way all users operate the site or based on what content is more popular than others, that way we are only seeing the content we are truly interested in.

The site should have a minimalist, simple, and clean look. Simplicity will add to the experience by making it easier for me to navigate. This way when I’m using the site I will be able to find what they are looking for that much easier. The website should be visually appealing so that I will be more likely to use the website for longer periods of time and more frequently. Using soft colors with simple text that has emphasis on key highlights should be easy on the eyes while still being visually stimulating without overloading with information. The content for the website should be displayed in a way that encourages me to want to watch the next next video or read the next article. Overall, the site should feel modern, bright, and straightforward. The color scheme should be consistent with company branding that way, I am reminded about the nature of the content I am seeing throughout my online experience. The website should have a simple drop down menu that is responsive so that I may navigate away if I need to but otherwise have a continuous stream of content on one page. Buttons or links should not be in the way of the content and training courses on the site.

The site will be maintained by the owner of the site. With the help of a user manual continued maintenance should be pretty easy. Training and a straightforward guide will explain how the site owner can upload new content with the same consistency as the web site’s original content. Other forms of maintenance includes logging into the WordPress account and adding photos, text, or other media to the website in a specified and labeled location where it will be hosted. The customer will also be able to maintain a record of email addresses of their user base in order to send updates informing the users of the website of any new and interesting changes. The maintenance of the site should be simple and changes should be swift and easy to make based on the friendly layout. The site will not require any major overhauls and it will be easy for the customer to understand with proper labelling what parts of the HTML and CSS code is doing and how they can interact with certain attributes to make light changes if needed. Updates to WordPress and various plugins will be as easy as clicking the button. Training on the importance of using secure passwords and backing up the site regularly will also be made available.

In conclusion, after many years of experience in parental guidance and providing therapy to children from a multitude of different backgrounds, I have decided that I can help many more people by building a platform that everyone can have access to. A website that gives parents the knowledge they need to talk with their children about difficult topics will be invaluable in this modern age. Some of the inspiration for this site comes from other online learning platforms like masterclass.com. These sites provide the end user with easy to digest materials through video, text, and imagery. Accessibility is so important for the success of these sites. The website has to be made easily available, content should be easy to pick up and come back to. Additionally, the content needs to be easy to understand, if only certain types of people can use it then it is pointless. The site itself needs to function smoothly, too much information is often to downfall of a website like this. Furthermore, the end user’s privacy is of utmost importance. In this regard, the site will only collect the user’s email address for maintenance reasons and also to notify them of new content. The core functionality of the site should be very similar on all types of devices. The look and feel of the site needs to be simple, clean, and minimalist. Light colors and nothing intense will benefit the end user’s viewing experience since they will most likely be viewing the site for long periods of time. The maintenance of the site should be able to be performed easily by the site owner. Being able to easily and quickly manage the site is important so it does not become stale. The WordPress platform allows for easily updatable content, plugins, and features. The website owner will be provided training materials on how to upload videos, articles, and images. In the end, the site’s ultimate goal is to provide easy and informative documentation to parents in a clean and straightforward fashion.

**b.Glossary of Terms**

WordPress: WordPress is a content management system written in PHP paired with a MySQL, MariaDB, or SQLite database.

**Functional Requirements Specification**

1. Stakeholders

(Nicole Richardson, Hieu Vu)

1. Actors and Goals

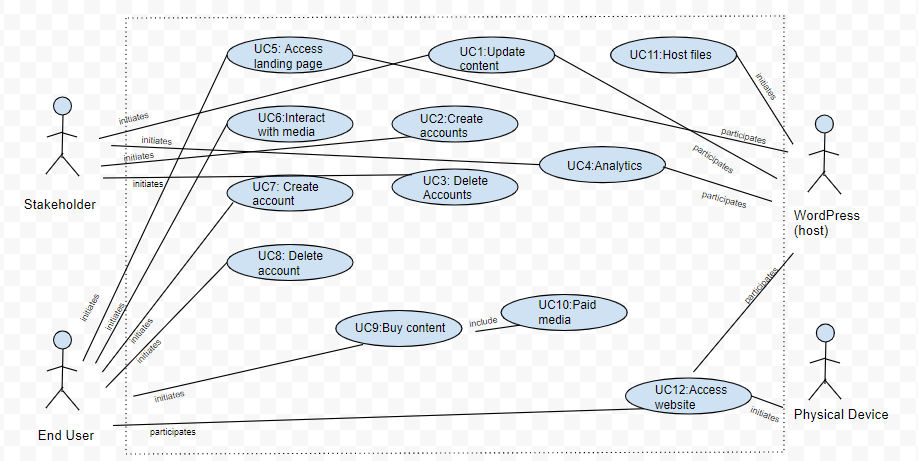
|  |  |  |
| --- | --- | --- |
| **Actor** | **Actor’s Goals** | **Use Case Name** |
| Stakeholder | To upload media content to website. | UC-1 |
| Stakeholder and End User | To create a user account. | UC-2 |
| Stakeholder and End User | To delete a user account. | UC-3 |
| Stakeholder | To access site analytics. | UC-4 |
| End User | To visit website landing page. | UC-5 |
| End User | To interact with media. | UC-6 |
| End User | To create an account. | UC-7 |
| End User | To delete an account. | UC-8 |
| End User | To buy additional content. | UC-9 |
| End User | To interact with paid media. | UC-10 |
| WordPress | To host website files. | UC-11 |
| Desktop/Mobile Device | To access website. | UC-12 |

c.Use Cases

An end user wants to be able to access the website. When accessing the website the end user may view a limited amount of free content, create an account, and purchase paid content. After the end user has purchased the paid content then they’ll be able to access the members only content. No matter if the user has purchased content or not they will be able to delete their existing account. Each user who creates an account on the site will have the ability to view their individual profile which gives them the option to remove their data/account from the website.

A stakeholder should be able to upload new media to the website. Additionally the stakeholder will be able to access their web files using WordPress. With these web files the stakeholder will be able create or delete any user accounts, and see website analytics. Website analytics will be powered by Google Analytics and interface with the website. WordPress, a third party system that will interact with the website and be the platform for all other interactions. WordPress will host the site and all the files created by the stakeholder and end users. WordPress will allow the stakeholders, who have authorized access, to update or make changes to the website. A desktop or mobile device are physical devices that will be the main conduit for interaction with the website. These devices will be able to view the website and allow the user to interact with the web site’s content. Which includes creating their account, deleting their account, and purchasing content.

Use Case Diagram



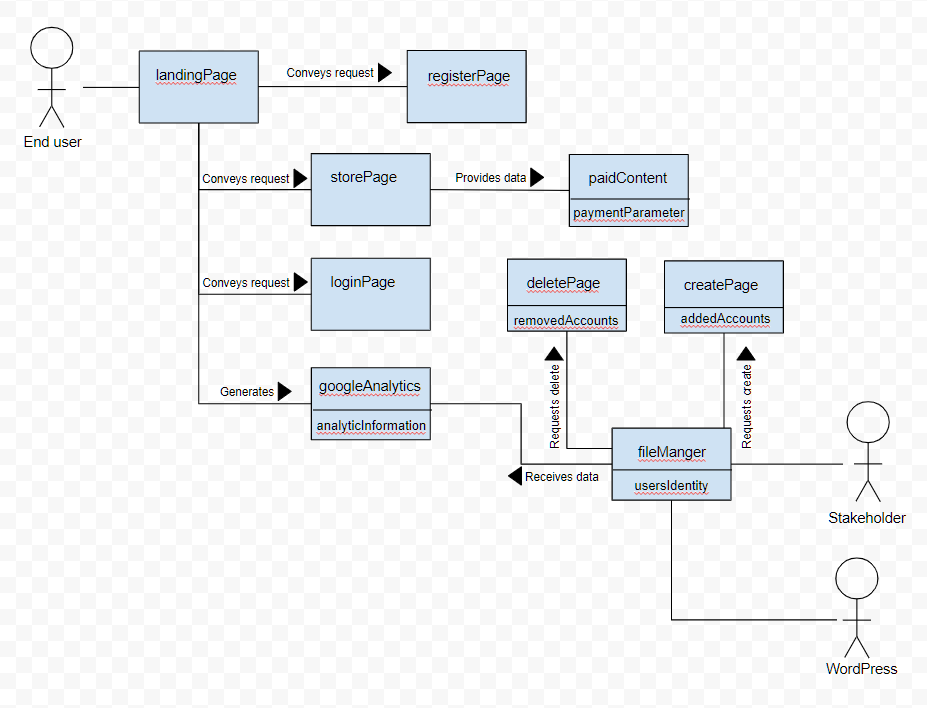
**Use case points**

|  |  |
| --- | --- |
|  | **Codework Digital** |
| **UUCP** | 52 |
| **TCF** | 1 |
| **UCP = UUCP x TCF** | 52 |
| **Normalized UCP** |  |

**Domain Analysis**

Domain analysis consists of identifying concepts beginning with boundary concepts (interacting directly with actors) and then moving inward to internal concepts, thus identifying the noun based objects that will need to be created and designing each object’s relationship to one another.

1. **Domain Model**



**b.Concept definitions**

|  |  |
| --- | --- |
| **Responsibility Description** | **Concept Name** |
| Store free content accessible to End Users (primary actor) with or without an account. | landingPage |
| Store optional content for purchase for End Users (primary actor) that have created an account. | storePage |
| Store login name and password submission fields with “login” button. | loginPage |
| Store login name and password submission fields to individual account creation with “create account” button. | registerPage |
| Store “delete account” button. | deletePage |
| Store “create account” button. | createPage |
| Store paid content accessible to End Users (primary actor) that have registered an account and are logged in. | paidContent |
| Store all HTML, CSS, PHP, and media files that can be accessed by End Users and updated by Stakeholders. | fileManager (host) |
| Integrated analytics through WordPress (host) accessible only by Stakeholder (primary actor) | googleAnalytics |

**c.Association definitions**

|  |  |  |
| --- | --- | --- |
| **Concept Pair** | **Association Description** | **Association Name** |
| landingPage <-> loginPage | landingPage allows the user to click on loginPage button. | Conveys requests |
| landingPage <-> registerPage | landingPage allows the user to click on the registerPage button. | Conveys requests |
| landingPage <-> storePage | landingPage allows the user to go to the storePage section. | Conveys requests |
| storePage <-> paidContent | storePage passes information to allow the user to access paidContent. | Provides data |
| fileManger <-> deletePage | fileManger generates the ability to use the deletePage feature. | Requests delete |
| fileManger <-> createPage | fileManger generates the ability to use the createPage feature. | Requests create |
| googleAnalytics <-> landingPage | Google Analytics passes on the information generated from the landingPage | Generates |
| fileManager <-> googleAnalytics | fileManager receives information about Google Analytics | Receives data |

**d.Attribute definitions**

|  |  |  |
| --- | --- | --- |
| **Concept** | **Attributes** | **Attribute Definition** |
| fileManager | userIdentity | Used to determine what content the user has access to. |
| paidContent | paymentParameter | Used to send payment information to payment processors. |
| deletePage | removedAccounts | Allows the ability to see what accounts have been deleted. |
| createPage | addedAccounts | Allows the ability to see what accounts are currently active. |
| googleAnalytics | analyticInformation | Traffic information is captured and sent to Google analytics. |

**Traceability matrix**

|  |
| --- |
| **Domain Concepts** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case** | **PW** | **landingPage** | **storePage** | **loginPage** | **registerPage** | **deletePage** | **createPage** | **paid**  **Content** | **file**  **Manager** | **Google**  **Analytics** |
| **UC1** | **4** | **x** | **x** |  |  |  |  | **x** | **x** |  |
| **UC2** | **3** |  |  | **x** | **x** |  | **x** |  |  |  |
| **UC3** | **2** |  |  | **x** |  | **x** |  |  |  |  |
| **UC4** | **2** |  |  |  |  |  |  |  | **x** | **x** |
| **UC5** | **6** | **x** | **x** | **x** | **x** |  |  |  | **x** | **x** |
| **UC6** | **3** | **x** |  |  |  |  |  | **x** | **x** |  |
| **UC7** | **3** |  |  | **x** | **x** |  | **x** |  |  |  |
| **UC8** | **2** |  |  | **x** |  | **x** |  |  |  |  |
| **UC9** | **4** |  | **x** | **x** | **x** |  | **x** |  |  |  |
| **UC10** | **5** |  | **x** | **x** | **x** |  | **x** | **x** |  |  |
| **UC11** | **9** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |
| **UC12** | **9** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |

**System Operation Contracts**

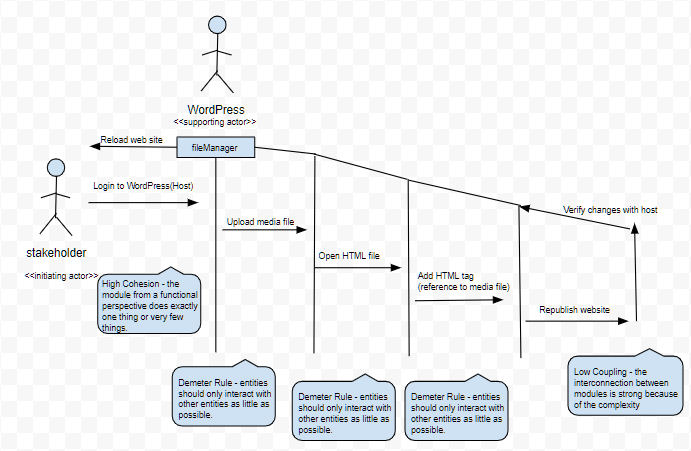
|  |  |
| --- | --- |
| **Operation** | Access landing page |
| **Preconditions** | Physical device with web browser, correct URL address, site hosted with WordPress |
| **Postconditions** | Landing page loads responsively in the device browser. |

|  |  |
| --- | --- |
| **Operation** | Buy additional content |
| **Preconditions** | Access landingPage, registerAccount, createAccount, storePage, select items to purchase, checkout with valid payment. |
| **Postconditions** | paidContent is now accessible |

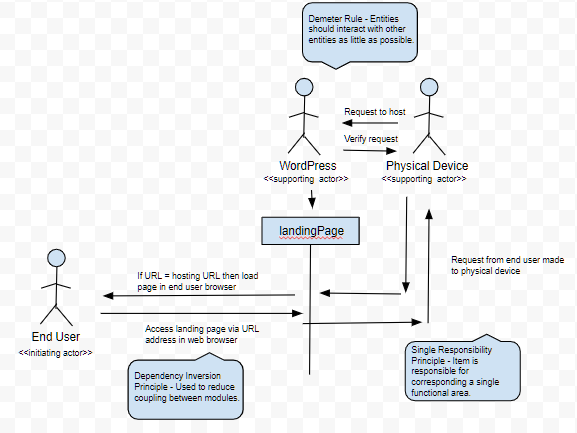
|  |  |
| --- | --- |
| **Operation** | Upload media content |
| **Preconditions** | Login to WordPress successful, HTML tag created, upload file into fileManager. |
| **Postconditions** | Newly uploaded media loads in the browser. |

**Interaction Diagrams**

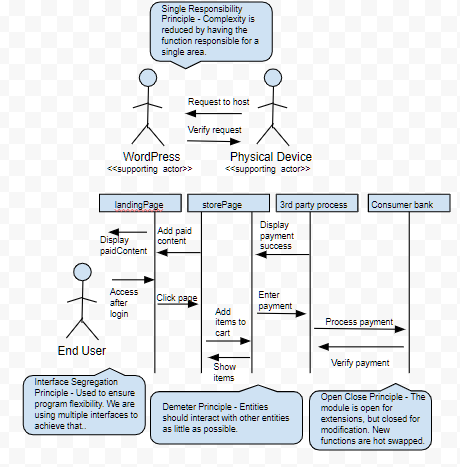
UC-1

****

UC-5

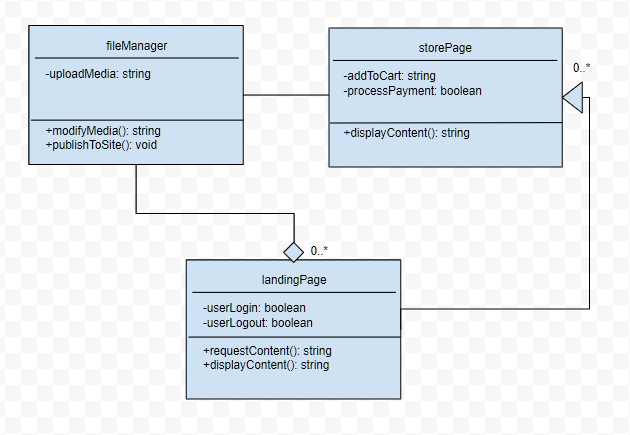
****

UC-9

****

**Class Diagram and Interface Specification**

* Class Diagram



* Data Types and Operation Signatures

***fileManager:***Class helps manage files on the WordPress site by allowing media to be uploaded to the site for various uses.

***uploadMedia*:** Attribute is private and is a string value. The value is correlated with the piece of media.

***modifyMedia*:** Operation is public and enables the user to modify the media to their liking. The media then returns a new string to update the old media with.

***publishToSite*:** Operation is public and is the piece that actually pushes the updated content to the site.

***landingPage:***Class allows users to view free content for subscription advertisement purposes. This page also gives the ability for users to create accounts, login and logout in order to access subscriber content.

***userLogin:*** Attribute is private and is a boolean value. This allows the user to login to the site.

***userLogout:***Attribute is private and is a boolean value. This allows the user to logout when finished accessing the site.

***requestContent:***Operation is public and lets the user request different versions of the landing page to be displayed.

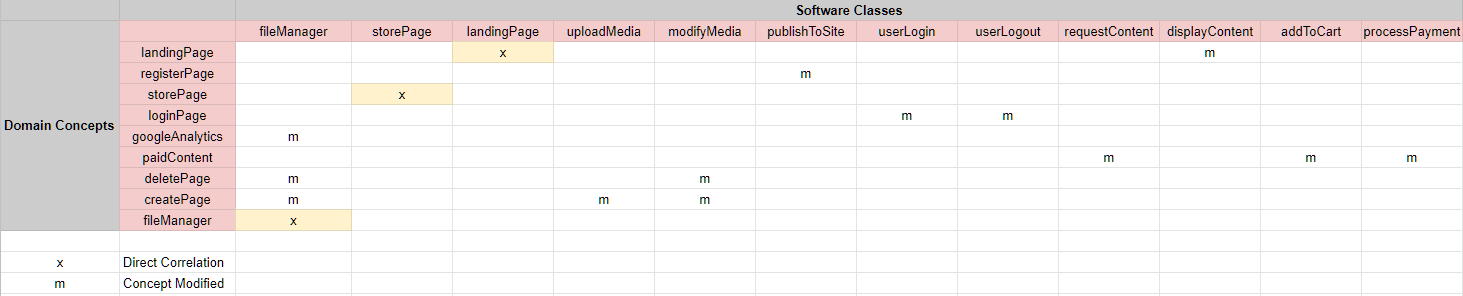
***displayContent:*** Operation is public and is the piece that actually pushes the content to be displayed.

***storePage:*** Class deals with the store page. This page allows users to purchase content that’s locked behind a paywall.

***addToCart:***Attribute is private and is a string value. The value is added to the backend to keep track of what the user is wanting to purchase.

***processPayment:***Attribute is private and is a boolean value. This tells the system if the user has decided to purchase the content.

* Traceability Matrix



The fileManager class evolved from googleAnalytics, deletePage, and createPage. The class was modified to allow for greater control in a single class.

The uploadMedia attribute was created out of the createPage class from the Domain Model. The uploadMedia attribute is a component in the fileManager class.

The modifyMedia operation was modified from deletePage and createPage class from the Domain Model. The modifyMedia operation is also a component in the fileManager class.

publishToSite is a more straightforward version of the registerPage class from the Domain Model. This was created to make its use more clear.

userLogin is an attribute that was created out of the loginPage function from the domain model. This was created so that it could function in the landingPage class.

userLogout is an attribute that was created out of the loginPage function from the Domain Model. This was created so that it could function in the landingPage class.

requestContent is an operation that stems from the paidContent function from the Domain Model. This feature was modified so that it could be apart of other functions rather than just for paid content.

displayContent similar to requestContent, this operation was created so that it could be utilized in multiple instances instead of a singular operation. It was modified from the landingPage in the Domain Model.

addToCart is an attribute that was created out of the necessity of something to push along the paidContent operation from the Domain Model.

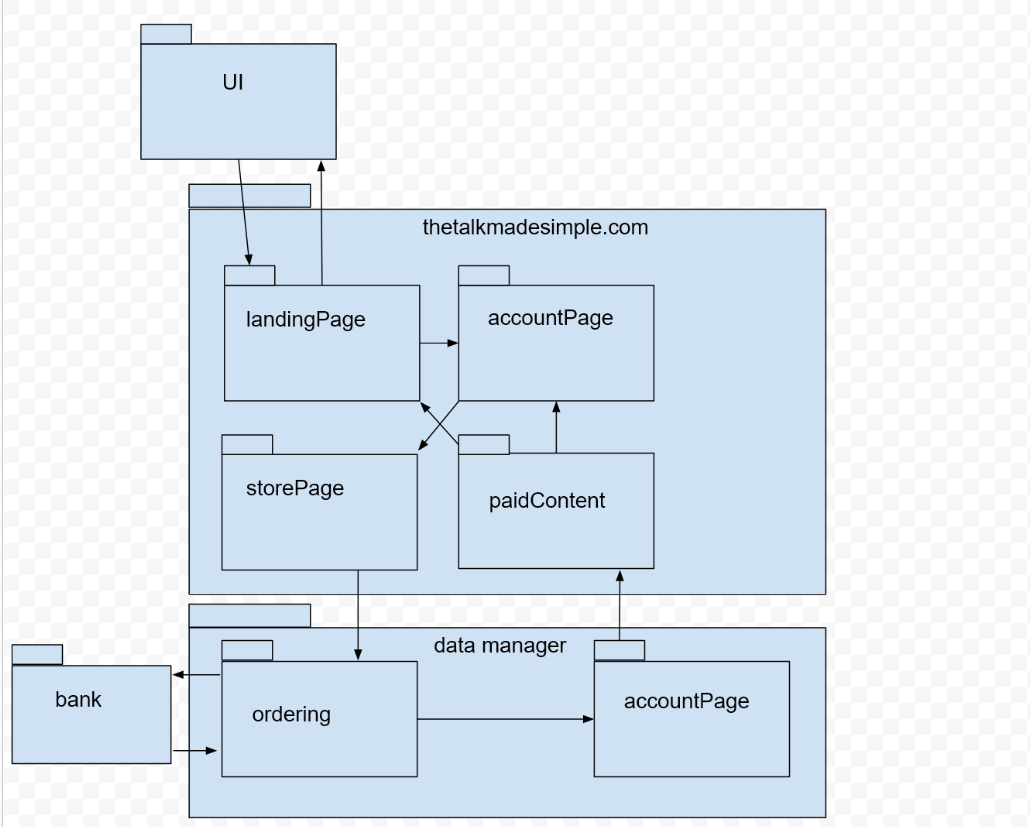
processPayment is another piece needed in order to fully realize the paidContent feature from the Domain Model.

**System Architecture and System Design**

* **Architectural Styles**

In our system which utilizes WordPress, the architectural style is closest to the Client-Server pattern. The pattern has two parties, the server and a single client or multiple clients. The server piece of the architecture which is WordPress provides services to multiple clients. The requests from clients are filtered through the server to retrieve the required end result. The server will continually listen for requests from the clients.

* **Identifying Subsystems**



* **Mapping Subsystems to Hardware**

The system really only needs a web server to function. However, to complete the program user’s must access it via a client workstation. The system is a website.

* **Persistent Data Storage**

The system will need to save some client data onto the web server. This data will be stored in the WordPress database. WordPress utilizes MySQL which is an open source relational database. Some of the data that will need to be saved are User Accounts, Content that’s been paid for, Analytics, Content Creation, and Notifications. WordPress is installed with 11 database tables by default. Those tables are…

1. wp\_posts
2. wp\_postmeta
3. wp\_options
4. wp\_users
5. wp\_usermeta
6. wp\_term\_taxonomy
7. wp\_terms
8. wp\_terms\_relationships
9. wp\_links
10. wp\_comments
11. wp\_commentmeta

* **Network Protocol**

Our system/site will utilize the HTTP and HTTPS protocols. These are the standard ports used to communicate with the web. HTTPS is used for security while HTTP is maintained for compatibility.

Mozilla describes HTTP and HTTPS as…

***HTTP*** *is a* [*protocol*](https://developer.mozilla.org/en-US/docs/Glossary/protocol) *which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance text, layout description, images, videos, scripts, and more.*

***HTTPS*** *(HTTP Secure) is an encrypted version of the* [*HTTP*](https://developer.mozilla.org/en-US/docs/Glossary/HTTP) *protocol. It usually uses* [*SSL*](https://developer.mozilla.org/en-US/docs/Glossary/SSL) *or* [*TLS*](https://developer.mozilla.org/en-US/docs/Glossary/TLS) *to encrypt all communication between a client and a server. This secure connection allows clients to safely exchange sensitive data with a server, for example for banking activities or online shopping.*

* **Global Control Flow**

Our website’s execution order is event-driven. Each user can choose which link to click on which will take them to various parts of the website. As for the time dependency our system is based on event-response. The system does not have a real concern for real time. When events are triggered the system will provide data. The WordPress content management system does not support multiple threads, so our website will not support it either.

* **Hardware Requirements**

Monitor Resolution: 1024 x 768 or higher

Minimum of 2 GB or RAM

An Internet Connection. Recommended speed 2Mbps or higher.

Keyboard and Mouse

Software Requirements

Modern Web Browser such as Google Chrome, Mozilla Firefox, Edge, Safari

* **User Interface Design and Implementation**

As of now we are still modifying and tweaking the final design of the site will be probably until the end of the project. We had a good idea of what we wanted the site to look like in the beginning which gave us a good baseline. In order to keep the client happy and adapt with flow of the work we will be making design changes as the project progresses. One of the main changes thus far is the change in the look of the course layout. Previously, the course layout showed the course steps on the top of the page around the header area. Now the course steps are going to be shown on the right side of the page. This will give the customer more of a sense of progression as they move down the list of requirements to complete the lesson. Other changes are minor things like color changes, shape changes, and imagery. The main functionality of the design is in place which includes the upload features, course creation, payment, admin, etc. Other user interface design layout changes we are making will be to instead have individual courses available for purchase; there will be an entire portion of the site that users can subscribe to in order to open up all paid content simultaneously. So this will change the look of the site so that individual courses are not shown or purchasable from the landing page, but rather the client can easily upload free individual videos to and the users can pay to subscribe to all site content which will then open up individual media and courses that will not require payment once the subscriber logs in.

* **Design of Tests**

Listed below is each case of our website’s core function that is expected to be working once the site has been launched:

1. Publishing content to the site via host (i.e Wordpress).
2. Content visibility on all pages.
3. Content visibility on paid account pages.
4. Creating user accounts.
5. Login with a user account.
6. Delete a user account.
7. Webpage can be navigated with links.
8. Webpage forms can be filled in and have default fields.
9. The website is consistent on both desktop and mobile device platforms.
10. All appropriate media content on the website is visible on both desktop and media platforms.
11. All cookies are encrypted before being stored on the user’s machine.
12. The website encryption security verification is working in all browsers.
13. Websites initial free content is visible on both paid and free account types.
14. Paid content is only visible on paid account types.
15. Payment and banking information for users are not stored on the website.
16. Payment transactions are encrypted.
17. Contact page sends message to appropriate recipient.
18. Able to create discount codes for paid content.

Listed below is each category of tests that will be performed on our website, these include; functionality tests, usability tests, interface tests, and compatibility tests. Below each category is the complete test coverage included in each test type.

**Functionality Tests**

**Links:**

1. Test all outgoing links on each web page.
2. Test all internal links on each web page.
3. Test to ensure there are not unreachable pages.
4. Test all links to external data hosts (i.e. email links, account creation links, payment links, sign in / sign out links)

**Forms:**

Test all forms on each webpage. (i.e. account creation form and payment information form.)

1. Test each validation field on forms.
2. Check that all default values in fields are correct.
3. Test that fields are correctly displaying incorrect inputs to the user.

**Cookies:**

Testing cookies by enabling and disabling cookies in all main browsers (i.e. mobile platform browsers, desktop browsers; Google Chrome, Firefox, Opera, Internet Explorer, Safari)

1. Check login sessions after ending sessions.
2. Check that cookies are encrypted before they are stored.

**HTML/CSS Validation:**

1. Check that website is searchable with all major search engines.
2. Validate HTML/CSS files using “W3C” markup validation service to check for errors.
3. Verify website encryption security service is working.

**Data:**

1. Verify that data entered into forms in account creation are accurately stored.
2. Verify that data stored in account creation are deleted upon request.
3. Verify that payment information entered in the validation field in forms is not stored after payment is completed.
4. Verify that payments are successfully completed.
5. Verify user access to paid/locked content is available once payment is recorded on account.
6. Verify that account status “paid/free” is accurately saved on each user account.
7. Verify that all accounts default to “free” account type.
8. Ensure customer personal information / data will not be shared with other party’s

**Useability Tests**

**Ease of Use:**

1. Verify that all elements on the website are visible on all desktop and mobile web browsing platforms.
2. Verify that all elements are displayed in the correct location to layout mapping.
3. Verify that all links are displayed and visible.
4. Verify that website appearance is accurate to layout mapping.
5. Verify the website color scheme and check for accuracy in design requirements and allow for ease of use in various accessibility user cases. (i.e. limited use on complimentary colors for color blind accessibility, etc.)
6. Verify that all website links and media links are easily accessible on mobile and desktop platforms.
7. Verify that the website can be quickly and easily navigated.
8. Check that all website media content is in the correct location on all webpages.
9. Check that all website form fields are accurately displayed on the correct pages.
10. Check that all website content is the correct size on both mobile and desktop browsing platforms.
11. Check that all media links and navigation links are properly labelled and allow for voice navigation for users with sight accessibility requirements.
12. Check for any text and spelling errors in all text based media on each page.
13. Verify that the site is easy to navigate and read in multiple browser “dark modes”.

**Interface Tests**

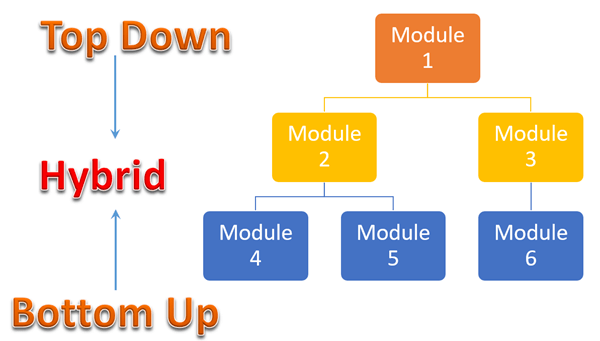
1. Verify that Google Analytics is implemented and functioning properly.
2. Verify that the payment system is functioning properly.
3. Verify that user accounts are defaulted and functioning.
4. Verify that the user's ability to create and delete their account is functioning.
5. Verify that account password security is functioning.
6. Verifying that incorrect validation fields are properly displaying error messages to the user.

**Compatibility Tests**

1. Verify website is consistent in all browsers on desktop and mobile platforms.
2. Verify that the website is compatible with all operating systems. (i.e Windows, Linux, Mac OS, Android, and IOS)
3. Verify that all media properly loads correctly and is properly autosized on all desktop and mobile platforms.
4. Verify that all forms are properly sized and visibly consistent on all mobile and desktop platforms.
5. Verify that any options to print site content is properly formatted on all desktop and mobile platforms.

**Integration testing**

For our integration testing we will be utilizing the incremental hybrid or “sandwich” approach.



In this strategy, testing is done by joining two or more modules that are logically related. Then the other related modules are added and tested for the proper functioning. The process continues until all the modules are joined and tested successfully. We will be using stub and drivers which are just dummy programs that test out our modules in order to test each individual one before connecting one module to another.

The advantages of the hybrid/sandwich strategy:

1. Allows parallel testing of various elements of the software system
2. Enables early testing of user interface components
3. Performs more coverage with the same stubs
4. It is a time saving process as several components are tested simultaneously

Upon adopting this test strategy; we will then begin to prepare our test data and cases. Using the interface and architecture design of our website we are able to map our critical modules for testing and map each use case that is logically related based on our test descriptions above.

**History of Work**

**Milestone/Deadline Evolution**

**Key Group Accomplishments**

* Organizing project communication via Slack and Trello.
* Organizing project reports and builds via Google Docs and GitHub.
* Meeting with Nicole Richardson and learning more about her ideas for her website.
* Creating a website through Godaddy hosting and WordPress content management system.
* Creating initiale theme template using Material Design Bootstrap and WordPress.
* Received progress feedback from client.
* Implementing and adjusting Analytics via Google Analytics.
* Implementing and adjusting notifications that customers are sent.
* Designing WordPress theme layout.
* Implementing WordPress menus.
* Implementing WordPress pages (Checkout, profile, landing page, etc.).

**Current Status**

Reviewing project status with stakeholder. Stakeholder is giving feedback and we are performing the modifications. Stakeholder wishes for all content to be restricted behind a paywall. Additionally, she wishes for the site to have all content available instead of broken up into courses. We are working on changing these details in the layout and in the payment structure for Demo 2.

**Future Work**

For future work we plan on creating training guides for the administrator in order for them to have an easier understanding on how to maneuver and edit their website whether its configuring a new account or deleting account emails. We also are planning to potentially add editable front end components which will help improve the website's overall personalization.

**References**

**GitHub:** GitHub is a web-based version-control and collaboration platform for software developers. Git is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers.

Link: <https://help.github.com/en>

**W3C Validation Service:** This website service is a free to use HTML/CSS file upload tool that checks these files for various logical and syntax errors, automatically.

Link: <https://jigsaw.w3.org/>