1. Introduction

1.1 The Problem Statement

Students face a challenge of always applying to internships and jobs. And employers waste their time interviewing unqualified students for the job.

Students find it time consuming and stressful when applying for jobs and internships as they have to deal with their already time consuming and stressful academics and keep up to date with latest job positions that are relevant to their studies or interests. Some – most don't keep up to date with the relevant job positions and rely on the ones that are advertised close by. This may lead to students missing out on opportunities that are relevant to them that they might be very interested in. Also, for the employer, they might have to settle for a less qualified or interested candidate.

1.2 Purpose

Student-Employer job match – is a web application which provides online application processing services for students and employers.

The users are final year students seeking employment and employers seeking graduates for potential jobs that the company has. It is an online service in which the students can use from the comfort of their space without having to travel for applications and submitting hard copy documents. This web application is quicker, easy and effective for the application process for graduates since students can

have very limited time to travel just to apply for jobs in their dream companies. This helps students to balance and not stress much on applying for jobs, and employers save time in processing hard copy cv's or student profiles.

1.3 Scope

- ✓ There are two basic users – Student and Employer.
- ✓ Both the student and the employer have their own profiles in **Student-Employer job match.**
- ✓ Students can make their online profile which will give the employer a detailed description about the student's studies and fields of study and interests so they can

  match a student with the desired job.
- ✓ Employers can also create their company's profile so that students can view that company and decide if it's a company they might interested in.
- ✓ Employers can schedule interview appointments to the student after the selection process.

1.4 Definitions, Acronyms and Abbreviations

XAMPP – Cross-Platform(X), Apache (A), MariaDB (M), PHP (P), and Perl (P).

PHP – Hypertext Preprocessor.

HTML – Hypertext Markup Language.

CSS – Cascading Style Sheets

1.5 Tools Used

XAMPP - is a free and open source **cross**-platform **web** server **solution** stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes

Adobe Muse is a website builder that allows designers to create fixed, fluid, and adaptive websites without having to write any code.[4] Muse generates static websites giving users the freedom to host their sites with any hosting provider. Users can add more advanced functionality such as blogging and eCommerce to their website with plugins created by third-party developers. This application is available through Adobe's Creative Cloud subscription. Muse will be discontinued as of March 2020, with the last feature improvements having appeared in March 2018.

Mobirise Website Builder is a freeware web design application, developed by Mobirise, that allows users to create and publish bootstrap websites, without coding.[2]Mobirise is essentially a drag and drop website builder, featuring various website themes. The software's features can be extended by purchasing extentions.

It has been featured in Huffington Post[2], IDG[3], TechRadar[4], About.com[5] and noted as an offline alternative to popular online website builders like Wix.com, Weebly, Jimdo, Webydo, Squarespace.

PhpStorm provides an editor for PHP, HTML and JavaScript with on-the-fly code analysis, error prevention and automated refactorings for PHP and JavaScript code. PhpStorm's code completion supports PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, and 7.2[2] (modern and legacy projects), including generators, coroutines, the finally keyword, list in foreach, namespaces, closures, traits and short array syntax. It includes a full-fledged SQL editor with editable query results.[3][4]

PhpStorm is built on IntelliJ IDEA, which is written in Java. Users can extend the IDE by installing plugins created for the IntelliJ Platform or write their own plugins.

All features available in WebStorm are included in PhpStorm,[5] which adds support for PHP and databases.[6] WebStorm ships with pre-installed JavaScript plugins (such as for Node.js).

it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

Apache - The **Apache HTTP Server**, colloquially called **Apache**, is a free and open-source **cross**-platform web server, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

The Apache HTTP Server is cross-platform; as of 1 June 2017 92% of Apache HTTPS Server copies run on Linux distributions.[5] Version 2.0 improved support for non-Unix operating systems such as Windows and OS/2. Old versions of Apache were ported to run on OpenVMS and NetWare.

Originally based on the NCSA HTTPd server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the World Wide Web, quickly overtaking NCSA HTTPd as the dominant HTTP server, and has remained most popular since April 1996. In 2009, it became the first web server software to serve more than 100 million websites. As of March 2018, it was estimated to serve 43% of all active websites and 37% of the top million websites.

MySQL - is an open-source **relational** database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter,[7] and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other

"AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google(though not for searches), Facebook, Twitter, Flickr, and YouTube.

PHP: Hypertext Preprocessor (or simply **PHP**) is a server-side scripting language designed for Web development, but also used as a general-purpose programming language. It was originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for *Personal Home Page*, but it now stands for the recursive acronym *PHP: Hypertext Preprocessor*

Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

JavaScript - often abbreviated as **JS**, is a high-level, interpreted **programming** language. **It** is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

1.6 References

- ✓ https://www.apachefriends.org/index.html
- ✓ https://en.wikipedia.org
- ✓ https://www.w3.org/html/
- ✓ https://www.apache.org/
- ✓ IEEE Recommended Practice for Software Requirements Specifications by Software Engineering Standards Committee
- ✓ https://docs.oracle.com/cd/E19528-01/819-4740/fwboh/index.html
- ✓ Software Requirements Specification, Bapuji Institute Of Engineering and Technology

1.7 Technologies to be used

- ✓ DB2: Relation Database Management System.
- ✓ Muse: **Adobe Muse** is a website builder that allows designers to create fixed, fluid, and adaptive websites without having to write any code.[4] Muse generates static websites giving users the freedom to host their sites with any hosting provider. Users can add more advanced functionality such as blogging and eCommerce to their website with plugins created by third-party developers. This application is available through Adobe's Creative Cloud subscription. Muse will be discontinued as of March 2020, with the last feature improvements having appeared in March 2018.[5][6]

Mobirise Website Builder is a freeware web design application, developed by Mobirise, that allows users to create and publish bootstrap websites, without coding.[2]Mobirise is essentially a drag and drop website builder, featuring various website themes. The software's features can be extended by purchasing extentions.

It has been featured in Huffington Post[2], IDG[3], TechRadar[4], About.com[5] and noted as an offline alternative to popular online website builders like Wix.com, Weebly, Jimdo, Webydo, Squarespace.[4][6]

PhpStorm provides an editor for PHP, HTML and JavaScript with on-the-fly code analysis, error prevention and automated refactorings for PHP and JavaScript code. PhpStorm's code completion supports PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, and 7.2[2] (modern and legacy projects), including generators, coroutines, the finally keyword, list in foreach, namespaces, closures, traits and short array syntax. It includes a full-fledged SQL editor with editable query results.[3][4]

PhpStorm is built on IntelliJ IDEA, which is written in Java. Users can extend the IDE by installing plugins created for the IntelliJ Platform or write their own plugins.

All features available in WebStorm are included in PhpStorm,[5] which adds support for PHP and databases.[6] WebStorm ships with pre-installed JavaScript plugins (such as for Node.js).[7]

Apache - The **Apache HTTP Server**, colloquially called **Apache**, is a free and open-source **cross**-platform web server, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

The Apache HTTP Server is cross-platform; as of 1 June 2017 92% of Apache HTTPS Server copies run on Linux distributions.[5] Version 2.0 improved support for non-Unix operating systems such as Windows and OS/2. Old versions of Apache were ported to run on OpenVMS and NetWare.

Originally based on the NCSA HTTPd server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the World Wide Web, quickly overtaking NCSA HTTPd as the dominant HTTP server, and has remained most popular since April 1996. In 2009, it became the first web server software to serve more than 100 million websites. As of March 2018, it was estimated to serve 43% of all active websites and 37% of the top million websites.

MySQL - is an open-source **relational** database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter,[7] and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other
"AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google(though not for searches), Facebook, Twitter, Flickr, and YouTube.

1.8 Overview

Existing System:

- ✓ Registration for the users both students and employers.
- ✓ Applying page for the students.

Drawbacks:

- ✓ Locally hosted for now.
- ✓ The system only uses only English.
- ✓ No web camera interaction to direct or conduct interviews.
- ✓ Students without internet connection cannot use the system.

Proposed System:

- ✓ Registration for students and employers.
- ✓ Website can suggest jobs for the students with regards to their field of study.
- ✓ Website can suggest employers with regards to the student's interest.
- ✓ Website will match the students with the employer that offers the jobs that the student wants.

Our Plan:

- ✓ Registration for students and employers (both users).
- ✓ Online matching of student and employer.
- ✓ Notification to student for potential matches.
- ✓ Setting up interviews online.

2. Interfaces

2.1 Interface Description

In computing, an interface is a shared boundary across which two or more separate components of a computer system exchange information. The exchange can be between software, computer hardware, peripheral devices, humans and combinations of these. Some computer hardware devices, such as a touchscreen, can both send and receive data through the interface, while others such as a mouse or microphone may only provide an interface to send data to a given system.

2.2 Software Interface

Client on Internet

Web Browser (any web browser), Operating System (any operating system)

Client on Intranet

Web Browser (any web browser), Operating System (any operating system)

Web Server

Apache, Operating System (any operating system)

Data Base Server

MySQL, Operating System (any operating system)

Development End

XAMPP (Apache, MySQL), myphpadmin, OS (Windows), HTML, PHP, JAVASCRIPT

2.3 Hardware Interface

Client Side

Browsers	Processor	RAM	Disk Space
Firefox Quantum 61.0.2	**Intel Pentium 4 or higher processor with SSE2 support**	**512MB**	**200MB**

Server Side

Technology	Processor	RAM	Disk Space
Apache	**Intel Pentium 4 or higher processor with SSE2 support**	**256MB**	**85MB**
XAMPP		**2GB**	**119Mb**

Recommended Requirements:

Client Side			
Browsers	**Processor**	**RAM**	**Disk Space**
Firefox Quantum 61.0.2	**Intel Pentium 4 or higher processor with SSE2 support**	**2GB**	**200MB**

Server Side			
Technology	**Processor**	**RAM**	**Disk Space**
Apache	**Intel Pentium 4 or higher processor with SSE2 support**	**2GB**	**100MB**
XAMPP		**2GB**	**119Mb**

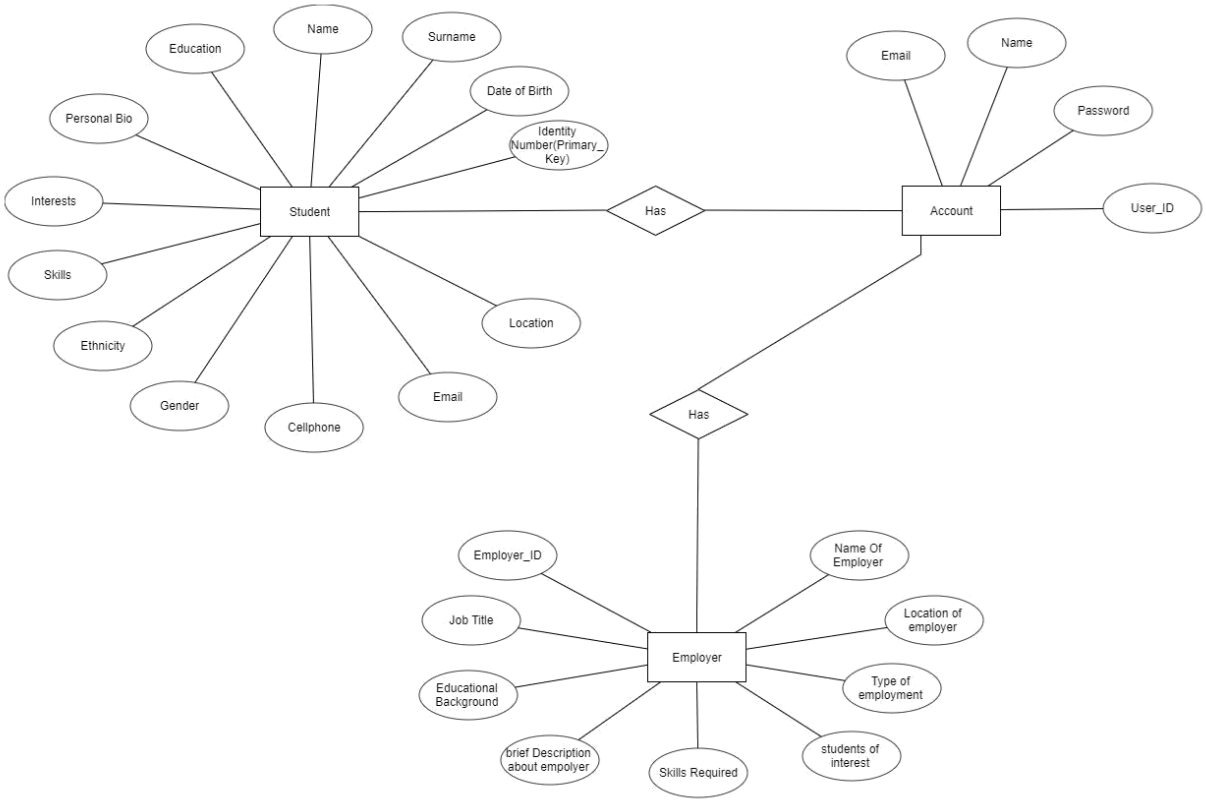
2.4 Communication Interface

- ✓ Client (student) on the internet will be using HTTP protocol.
- ✓ Client (employer) on the internet will be using HTTP protocol.

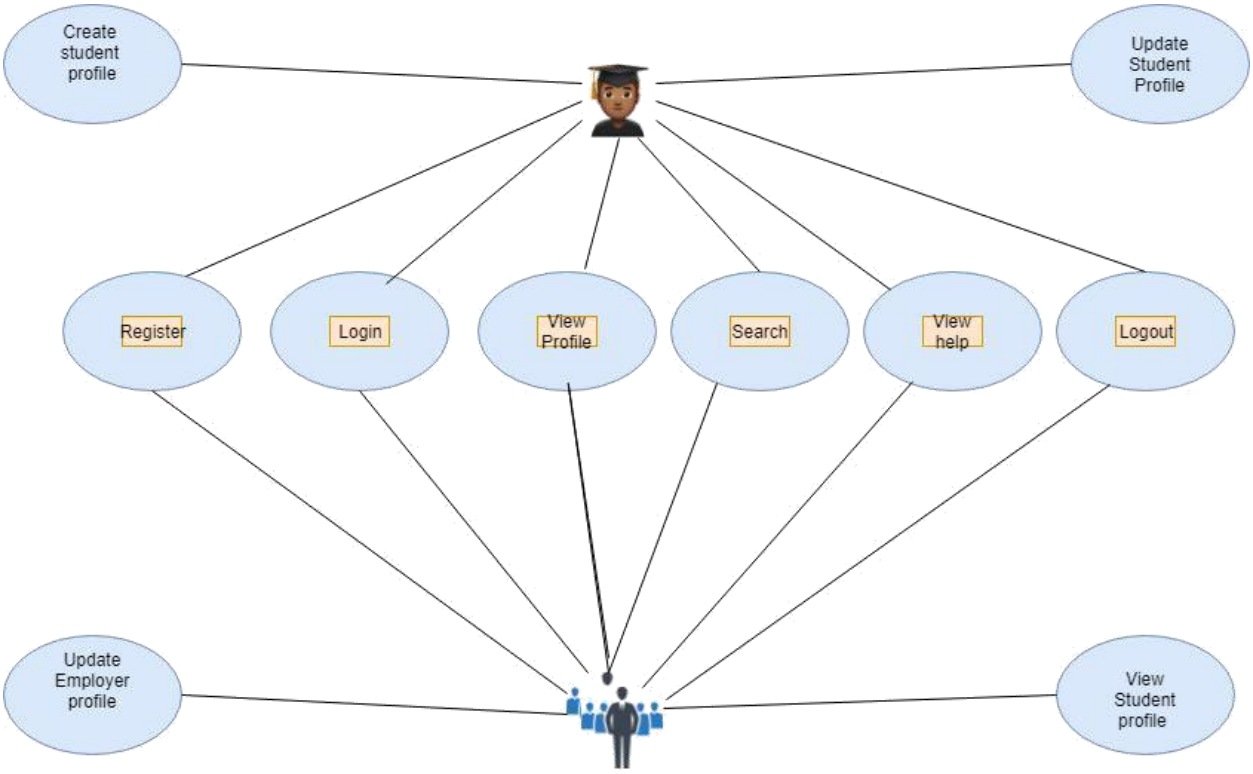
2.5 Constraints

- ✓ GUI is only in English.
- ✓ Login and password is used for the identification of the users.
- ✓ Only registered students and employers will be authorized to use the services.
- ✓ Limited to HTTP

2.6 ER Diagram



2.7 Use Case Model Survey



Student:

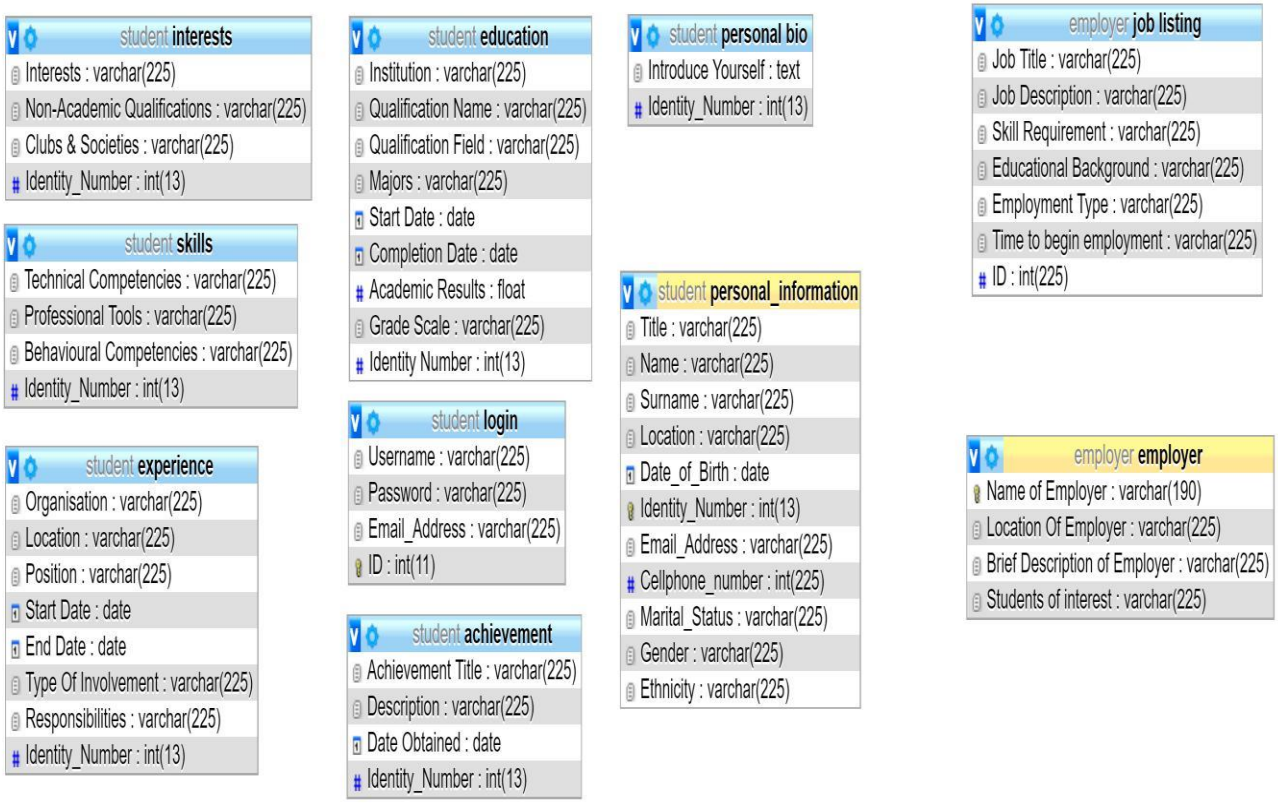
Students can create their profiles and update their profiles online. Students can also get help from the website and view profile of the employer that they are matched with.

Employer:

The employer can view student profile after the matching. They can update their profiles as to what jobs the company offers. They view the student's profiles in which they are matched with.

2.8 Architecture Diagram

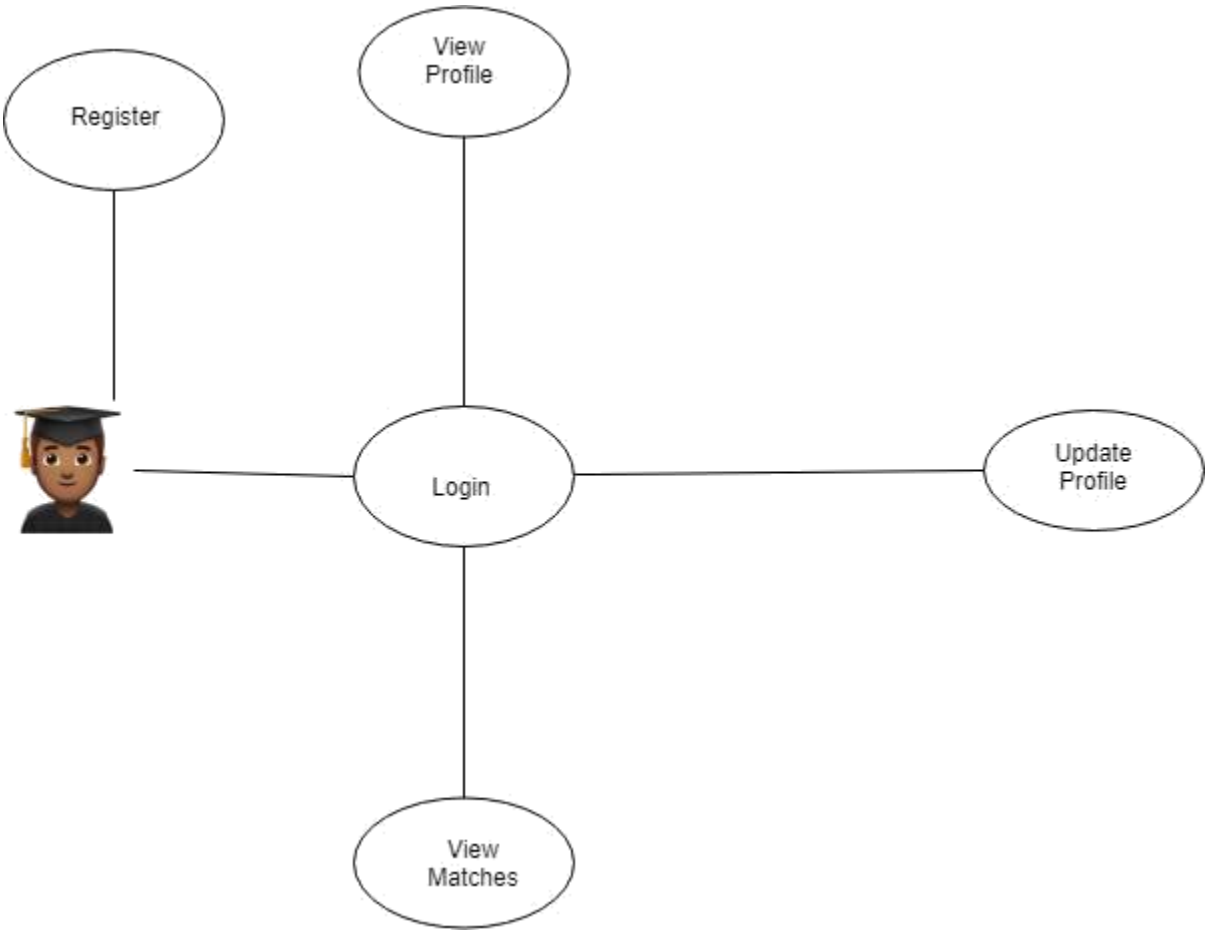
2.9 Database Design



3 Specific Requirements

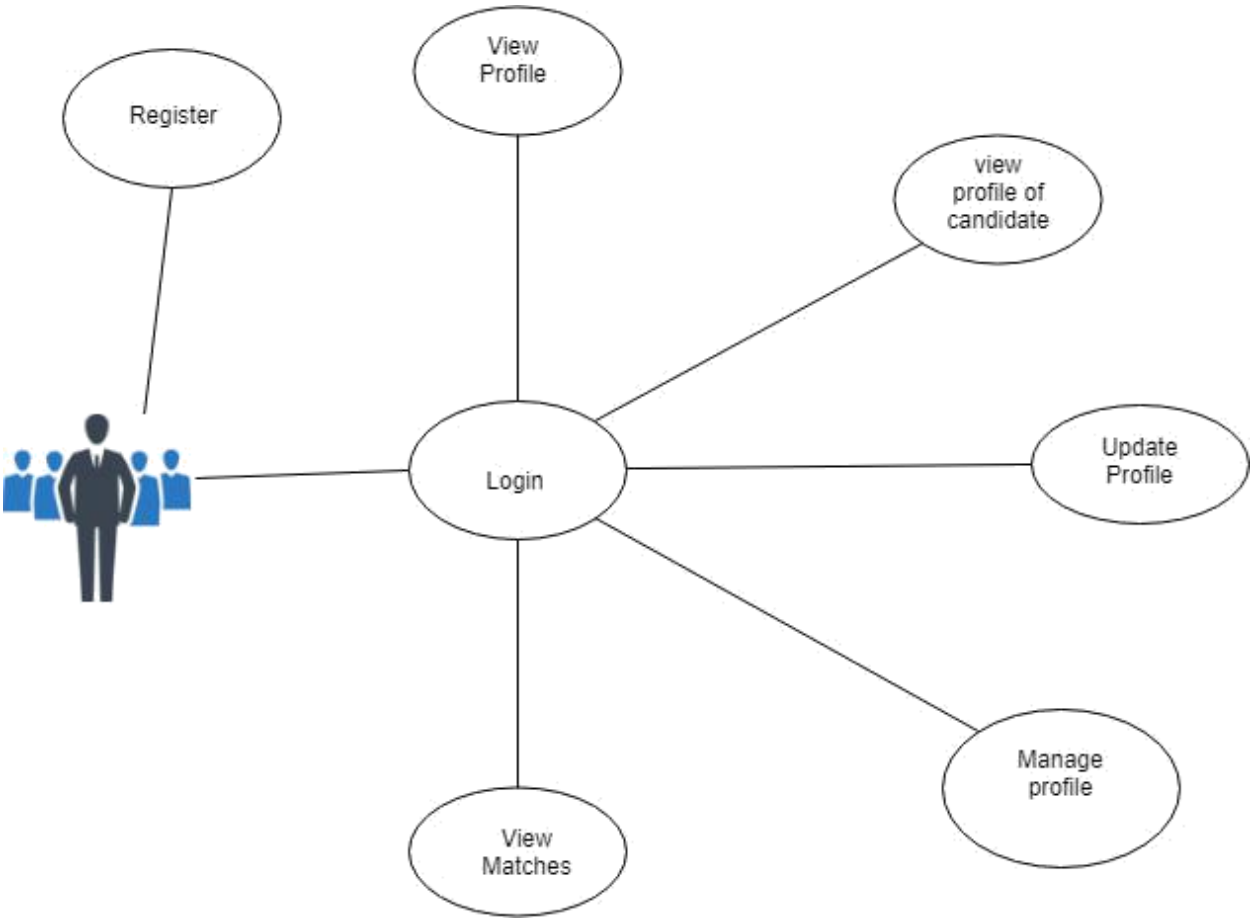
3.1 Use Case Reports

3.1.1 Student use-case report



Use Case	Description
Register	The student has to register in order to use the services provided by our app
Login	The student has to log in in order to use the below services
View Profile	This is the profile created by the student that contains the students personal details
View Matches	The student can view the jobs that she/he has been matched with

3.1.2 Employer use-case report



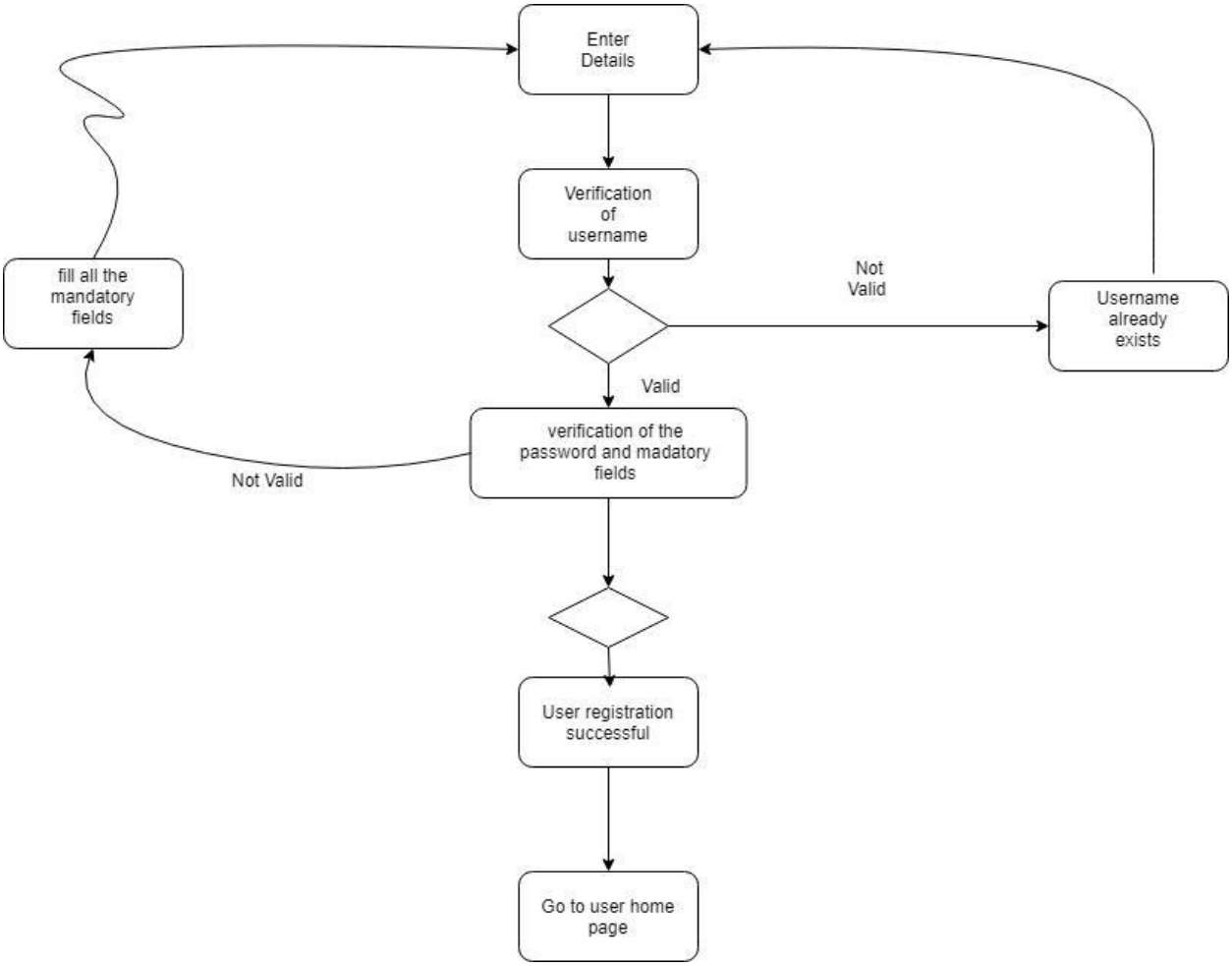
Use Case	Description
Register	**The employer has to register in order to use the service**
Login	**The employer has to login in order to look at potential students**
View Profile	**The employer can view their profile while logged in containing their professional details**
View profile of candidate	**The employer can view profile of students that match jobs advertised**
Update profile	**The employer can update the company's/employers profile**
Manage profile	**The employer can manage their profile by deleting the jobs which are already occupied or posting new jobs to be advertised**

3.2 Activity diagrams

3.2.1 User Registration

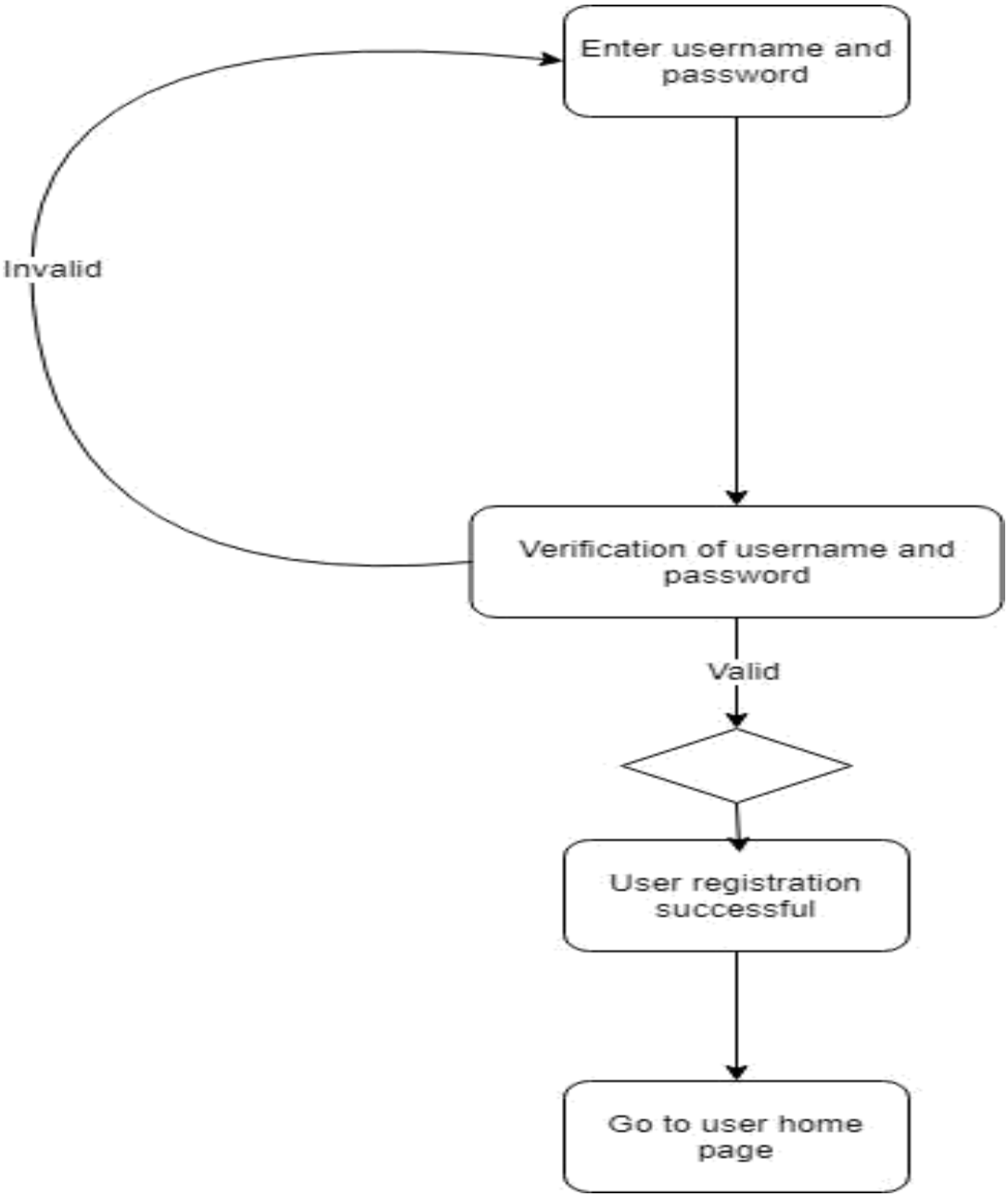
In this web application every user initially has to fill in the mandatory fields in order to be registered. Then once the user submits the mandatory details needed, his credentials are verified and added to the database. If the username already exists then the user is taken back

to choose another username. The fields cannot be left empty to register, the program will give an error, and the web app will prompt the user for the correct input.



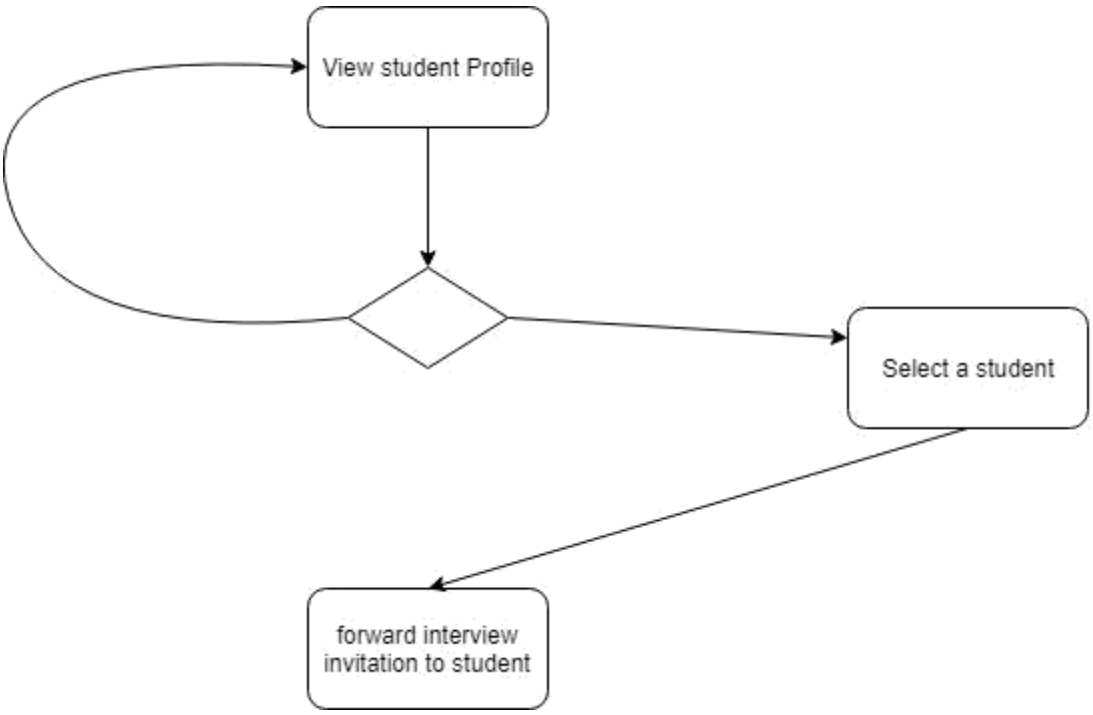
3.2.2 User Login

The user is made to enter the username and password used to register to log in. The values are verified. If the values are valid then the user is logged in or of the values are not valid they are asked to re-enter the correct values.



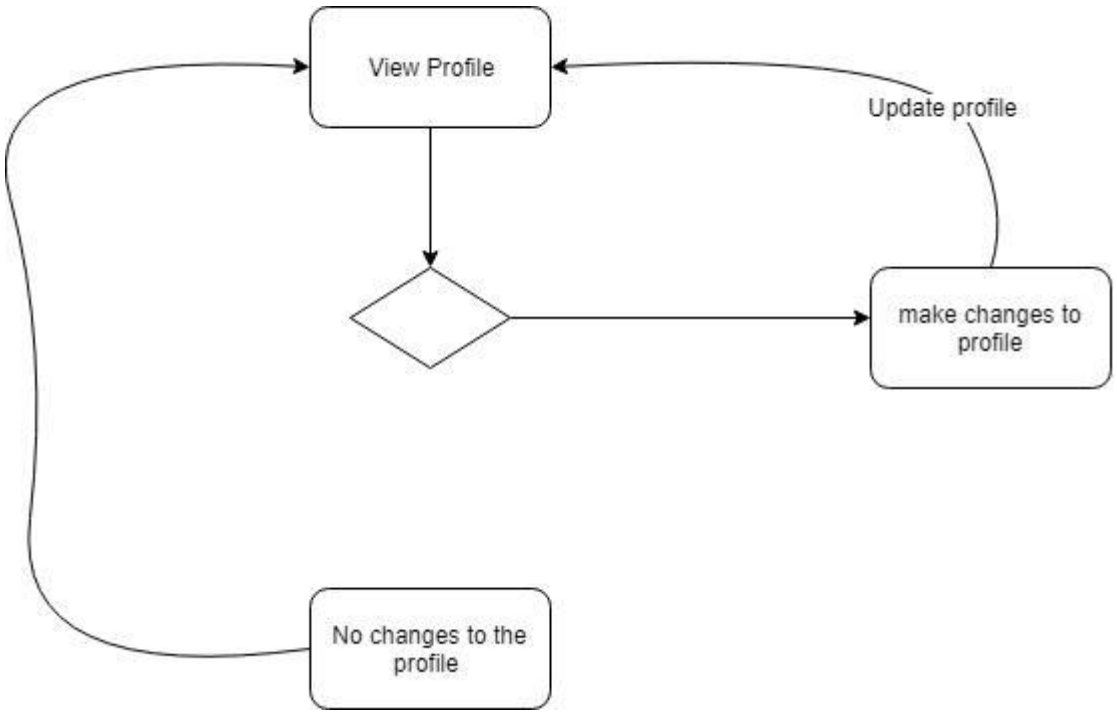
3.2.3 View Student Profile

The employer can want to view the students profile after the match. When the employer is impressed or the student meets the minimum requirements, the employer then forwards an interview invitation



3.2.4 Student Profile

The student might have new information about themselves. The student might want to update their profile, they might have details that they would love to add to their profiles, if not the profile remains the same.

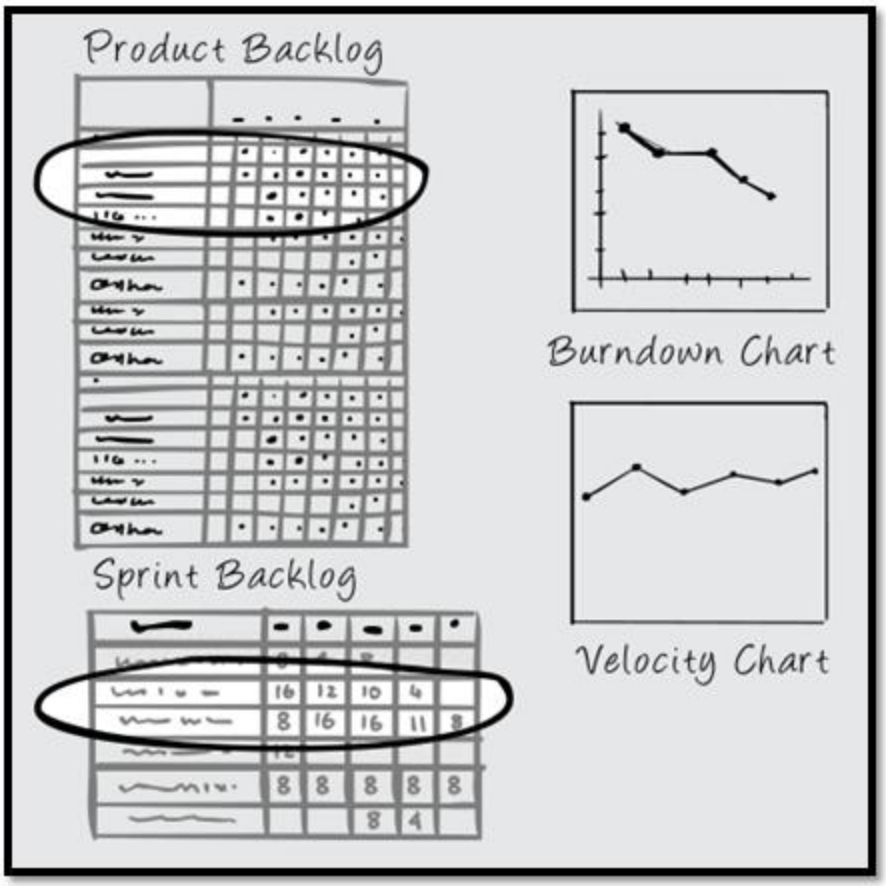


4. Sprint Planning Document

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box.
Sprint Planning answers the following:
What can be delivered in the Increment resulting from the upcoming Sprint?
How will the work needed to deliver the Increment be achieved?
Work is selected from the Product Backlog and pulled into the Sprint Backlog. Now remember that the work in the Sprint Backlog is not a commitment, it is a forecast. The only container of a Sprint is its time box, not the work planned for the Sprint. Learn more about the Sprint Backlog and Product Backlog.

Sprint Goal

The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. As the Development Team works, it does so with the Sprint Goal always in mind


Product Backlog
Sprint Backlog
16 12 10 4
8 16 16 11 8
8 8 8 8 8
8 4
Burndown Chart
Velocity Chart

WHAT IS SPRINT PLANNING?

Sprint planning is a timeboxed working session that lasts roughly 1 hour for every week of a sprint. In sprint planning, the entire team agrees to complete a set of product backlog items. This agreement defines the sprint backlog and is based on the team's velocity or capacity and the length of the sprint.

WHO DOES IT?

Sprint planning is a collaborative effort involving a ScrumMaster, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who define the work and effort necessary to meet their sprint commitment.

HOW DO WE PREPARE?

Ensure all sprint candidates meet the team's definition of *ready*. In the days and weeks leading up to sprint planning, the Product Owner identify the items with the greatest value and works towards getting them to a ready state.

- Assign a relative story point value
- Remove dependencies
- Create testable examples
- Define acceptance criteria
- Meets INVEST criteria

WHAT IS THE BACKLOG?

The product backlog can address just about anything, to include new functionality, bugs, and risks. Product backlog items (PBI's) must be small enough to complete during a sprint and should be small enough to complete within a few days. All stories must be verified that they are implemented to the satisfaction of the Product Owner.

ENSURE RIGHT SIZING BACKLOG ITEMS

Based on historical data of the team, first determine if product backlog items are too large to complete in a sprint. In these cases, do not consider these stories as valid sprint backlog candidates. Rather, in order to consider for sprint planning, split the stories into smaller pieces. Additionally, each story must be able to stand on its own as a vertical slice. Therefore, stories should not be incomplete or process-based as a horizontal slice.

CALCULATING A COMMITMENT

To calculate a commitment, mature teams may use a combination of both team availability and velocity. However, new teams may not know their velocity or they may not be stable enough to use velocity as a basis for sprint planning. In these cases, new teams may need to make forecasts based solely on the their capacity.

DETERMINING VELOCITY

First of all, as velocity is unique to every team, never use another team's velocity to plan your sprint. Derive team velocity by summing the story point estimates of all completed and accepted work from the previous sprint. By tracking team velocity over time, teams will begin to focus less on utilization and consequently more on throughput.

DETERMINING CAPACITY

For teams without a stable velocity, each team member should provide three simple measures to determine capacity. First, what are the number of ideal hours in their work day? Second, how many days in the sprint will that person be available? Third, what percentage of time will that person dedicate to this team?

THE PLANNING STEPS

1. Remind the team of the big picture or goal
2. Discuss any new information that may impact the plan
3. Present the velocity to be used for this release
4. Confirm team capacity
5. Confirm any currently known issues and concerns and record as appropriate
6. Review the definition of *DONE* and make any appropriate updates based on technology, skill, or team member changes since the last sprint
7. Present proposed product backlog items to consider for the sprint backlog
8. Determine the needs, sign up for work, and estimate the work owned
9. Product Owner answers clarifying questions and elaborates acceptance criteria
10. Confirm any new issues and concerns raised during meeting and record
11. Confirm any assumptions or dependencies discovered during planning and record
12. ScrumMaster calls for a group consensus on the plan
13. Team and Product Owner signal if this is the best plan they can make given what they know right now
14. Get back to work

4.1 Sprint 0

Work than needs to be done by Wednesday the 8th of August 2018

We basically do not want to bore our clients by making them fill in long forms, instead we take them step by step per page

1. The basic layout of the web. The prospective candidate's perspective (own database, first). Meaning what each page entails and how it looks.

Page 0 - Welcome/home page (no table)

a. Create account and specify type

Page 1 - Personal information (own table)

a. names (columns)
b. birth data and age (columns)
c. nationality (column)

Page 2 - Educational background (own table).

Page 3 - Skills and expertise (own table).

Page 4 - work experience (own table).

Page 5 - achievements (own table).

Page 6 - projects (own table).

Page 7 - interests (own table).

Page 8 - job interest titles (own table).

2. The basic layout of the web from the employer's perspective (own database, second). Meaning what each page entails and how it looks.

Page 0 - Welcome/home page (no table)

A. create account and specify type

Page 1 - Employee company information (own table)

a. Company name (column)
b. Location (column)
c. Areas of interest of specialization (column)

Page 2 - Job listings (own table)

Page 2a - Educational background required for this specified job (own 'sub-table' or even column).

Page 2b - Skills and expertise required for this specified job (own 'sub-table' or even column).

Page 2c - minimum work experience required for this specified job (own 'sub-table' or even column).

Page 2d - achievements advantageous for this specified job (own 'sub-table' or even column).

Page 2e - projects kinds advantageous for this specified job (own 'sub-table' or even column).

Page 2f - Personal information required (table)

a. Nationality
b. Age
c. Marital status
d. Identity number

Page 2g - Employment type (column)

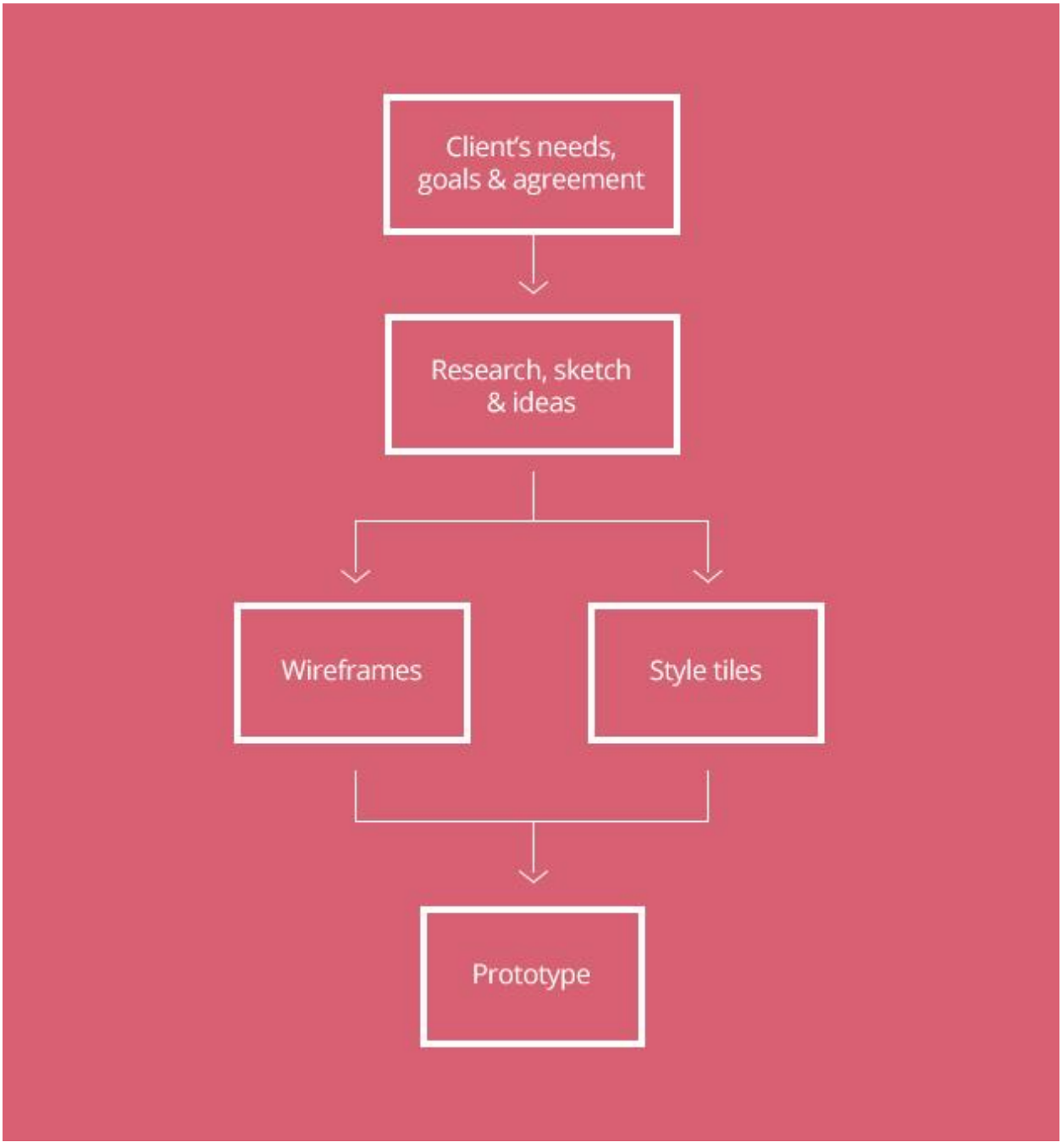
a. Permanent
b. Temporary
c. Contract

Page 2h - Job title (column)

Page 2i - Time for employment to begin (column)

Page 2j - interests (not actually necessary) (own 'sub-table').

4.2 Sprint 1


Client's needs, goals & agreement
Research, sketch & ideas
Wireframes
Style tiles
Prototype

One of the most important factors of the design industry today, especially freelance design, is a successful business workflow and doing more with less time. And to be time efficient, you must take the time to plan your project, by breaking it down to basic steps in order to achieve a successful outcome. This way, you improve productivity.

“Productivity is never an accident. It is always the result of a commitment to excellence, intelligent planning, and focused effort.” ~ Paul J. Meyer

Each designer has his own favorite design process. In this article I will explain my suggested time-efficient design workflow that works for me every time. This article is targeted at newcomers to the web design industry rather than experienced designers. The point of the article is to provide a general 5 step guide to a successful website design.

Step 1: Client's needs

Before starting your plan, you'll need as much information from your client as possible. The more you know, the more back-and-forth e-mails you'll save.

Listen to their ideas

Usually, clients won't tell you what they want, it's up to you to ask them. They want reassurance that their brief will be fulfilled the way they visualized it, but they have a hard time explaining how they see their future website. The best way to overcome this problem is communication. It's essential to get to know your website design clients by asking questions that will tell you about their business. You want to find out the mood they're going for, what they like and what they don't.

Client questionnaire

My favorite approach to find out about my client is by filling a form. List the most important questions that will help beginning your project. What's the purpose of this website? Who's the target audience? Who are your competitors? What actions do you want the visitors to take? Etc.. The form should not exceed 10 questions, because let's be realistic, not all clients have the time nor the patience to fill out 20 or 30 fields. Make sure to send the client your form before you start drafting the proposal and not after. If you want to see an example, you can check out this form. For accessibility reasons and ease of use, I recommend you set up your form on Google Drive Forms. Now that the client has filled your form, you can send your proposal, and finalize your formal agreement on budget, scope of project, timeline, and ownership. After signing the agreement, it's time to get to work.

Step 2: Research, ideas & sketch

This step is always the hardest. When I first started in the industry, I used to jump into Photoshop and start designing. I can't stress enough how wrong that is. The best thing to do is to start on a piece of paper.

Start sketching

If you're a designer and you don't have a personal notebook, you better drive to your nearest library. Writing down your notes using a pen and paper is the most efficient way when it comes to any activity that involves brainstorming. Take some time and start sketching any idea that pops to your head. You don't have to know how to draw. You can start with a circle, then your ideas will flow naturally. You can be in your taxi on your way to work, and a creative idea can pop. Write it down.

Compile your inspiration

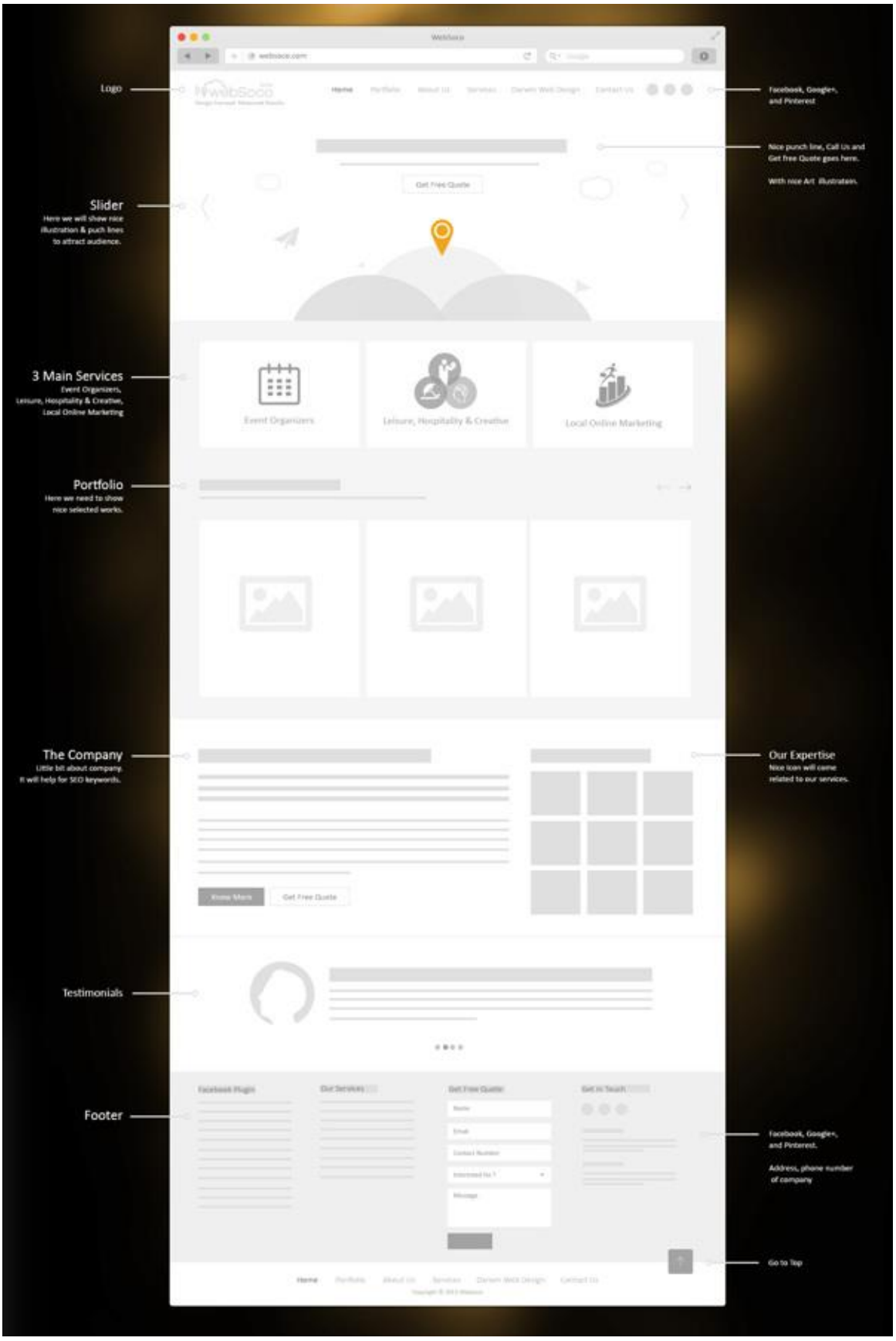
Create a bookmark folder with the project's name. Start researching and then save any webpage you deem relevant. Make sure not to copy exact elements or ideas as this will be considered plagiarizing. Find your best looking websites and layouts based on the client's needs and search for other websites in the same business. Write down what you like and what you don't like about them. Layouts, colors, shapes, animations.. Anything that will give you ideas and inspiration for your next step.

Know your competitors

Take a good look at your competitors' websites. Check all the local and international competitors and learn from their content. What do they all have in common? This is what you should include in your website. What are they missing? That's what you can improve in your website.

Step 3: Wireframes

Wireframes are the blueprint of your website. It's what gives your website a visual structure. Some designers have the tendency to include design elements in their wireframes. This should be avoided completely, as this will distract the purpose of the wireframes, which is to state where things go, not how things look like. Wireframes should be in grayscale to focus only on the layout. The first approach to a wireframe is, yes, a pen and paper. List all your elements to include on a web page. Next, arrange them on the page. You can do that also on paper or by using any wireframe tool such as proto.io, wireframe.cc or even Photoshop. It's a simple layout to start working from. Here's a nice example of a wireframe:


Logo
Facebook, Google+, and Pinterest
Nice punch line, Call Us and Get free Quote goes here.
With nice Art Illustraten.
Slider
Here we will show nice illustration & puch lines to attract audience.
3 Main Services
Event Organizers, Leisure, Hospitality & Creative, Local Online Marketing
Event Organizers
Leisure, Hospitality & Creative
Local Online Marketing
Portfolio
Here we need to show nice selected works.
The Company
Little bit about company. It will help for SEO keywords.
Our Expertise
Nice Icon will come related to our services.
Testimonials
Footer
Facebook, Google+, and Pinterest.
Address, phone number of company
Go to Top

Element placement

You can’t start your website without knowing where things go. Here’s where you decide that. You can even meet up with your client to discuss the best placement for your content.

Information hierarchy

A sitemap with a list of links can be overwhelming to a client and even the designer. By laying out the sitemap, you will be able to see how the pages fall into place which will allow you to make adjustments before you get too far ahead.

Interactivity

When creating your wireframes, you must ask your questions: How are these elements going to interact with each other? Will the visitor take the actions I want him to take? How will the visitor move around the site? Here’s where user experience plays its role. So you created your layout. Next, you will create your interface.

Step 4: Style tiles

Style tiles (also known as style guides) are a design deliverable consisting of fonts, colors and interface elements that communicate the essence of a visual brand for the web. Style tiles are based on visual preference discussions with the client The usual elements consist of the logo, typography, colors and tone of voice.

From concept to visual

Composing your style tiles will translate the mood, feeling and tone of your website based on what you learned from your client’s needs. It will help you

make choices about how to use the elements of design in a way that the client will understand.

Maintain consistency

With a style tile, you are able to maintain a look and feel by creating a set of rules which the design follows. The process becomes flexible, easily to maintain and consistent.

Time efficient

Designing your style tiles will show your client the website's skin. Instead of composing your full designed website in a certain style, now you can update your style tiles with the client's requests in a quick and efficient way. Now that you have designed your skin. Next, you will apply it.

Step 5: Prototype

Designing a website is very much like building a house; the floor plans are the wireframes, and the visual interior design is your style tiles. The wonderful thing about this design process is how easy this step has become. By composing your wireframe and style tiles, all you need to do now is combine them together. Here's where you start seeing how your website will look like. Prototyping means building a model of the website before you build the actual website. Some designers like to design their prototypes on Photoshop, while others like to skip Photoshop and move onto a web framework such as inVision and UXpin.

Prototyping is based on 3 steps:

Prototype

Implement the style tiles into the layout while adapting user experience and ensuring design ease-of-use.

Test

This step is essential as it evaluates whether the website meets the client's needs and expectations. You can even share your prototype with your users to test your interactivity and user experience.

Fine-tune

Here's where you refine any alterations and elements that need to be improved. This process may take time depending on how many cycles that you and the user/client will take in order to have a final polished design.

Finishing up

So now you should be able to have an idea about the best process for a successful website design. By following this workflow, you can now plan your website ahead, prevent surprises that can block your workflow, and most importantly avoid time-consuming projects. The next step will be the development part

4.3 Sprint 2

What is Mobirise?

Mobirise is a free **offline app** for Windows and Mac to easily create small/medium websites, landing pages, online resumes and portfolios, promo sites for apps, events, services and products**.**

Who is it for?

Mobirise is perfect **for non-techies** who are not familiar with the intricacies of web development and for designers who prefer to work as visually as possible, without fighting with code. Also great for pro-coders for fast prototyping and small customers' projects.

Why Mobirise?

Key differences from traditional website builders:

* Minimalistic, extremely **easy-to-use** interface
* **Mobile**-friendliness, latest website blocks and techniques "out-the-box"
* **Free** for commercial and non-profit use

HOW MOBIRISE WORKS?

Drag blocks to page

Start with creating a new website and picking up the theme. Then expand the blocks panel with the big red "plus" button in the lower right corner and start dragging the blocks you like.

Edit and style inline

Edit the content of each block just like you would in a regular text editor, click on media elements to insert your own image, video or icon.

Set block parameters

Set the appropriate options through the Parameters Panel being brought out by the blue "gear" button.

Preview and Publish

Click on "Mobile View" in the website builder to check how your site looks on different devices and publish to local drive, FTP or free Github Pages.

PhpStorm deeply understands your code.

Major frameworks supported

PhpStorm is perfect for working with Symfony, Drupal, WordPress, Zend Framework, Laravel, Magento, Joomla!, CakePHP, Yii, and other frameworks.

All PHP tools

The editor actually 'gets' your code and deeply understands its structure, supporting all PHP language features for modern and legacy projects. It provides the best code completion, refactorings, on-the-fly error prevention, and more.

Front-end technologies included

Make the most of the cutting edge front-end technologies, such as HTML5, CSS, Sass, Less, Stylus, CoffeeScript, TypeScript, Emmet, and JavaScript, with refactorings, debugging and unit testing available. See changes instantly in the browser thanks to Live Edit.

Built-in developer tools

Perform many routine tasks right from the IDE, thanks to Version Control Systems integration, support for remote deployment, databases/SQL, command-line tools, Vagrant, Composer, REST Client, and many other tools.

PhpStorm = WebStorm + PHP + DB/SQL

All the features of WebStorm are included into PhpStorm, and full-fledged support for PHP and Databases/SQL support are added on top

5. Sprint Retrospective

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is at most a three-hour meeting for one-month Sprints. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. This is the opportunity for the Scrum Team to improve and all member should be in attendance.

During the Sprint Retrospective, the team discusses:

- What went well in the Sprint
- What could be improved
- What will we commit to improve in the next Sprint

The Scrum Master encourages the Scrum Team to improve its development process and practices to make it more effective and enjoyable for the next Sprint. During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by improving work processes or adapting the definition of “Done” if appropriate and not in conflict with product or organizational standards.

By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

5.1 Sprint 0 Retrospective

Meeting Minutes 15/08/18

- Backend had a completed students database
- Backend was told to complete employee and job listing table
- Frontend a page drafts where started
- Next implement the forms to match the backend

Next Meeting Wed 22/08/18

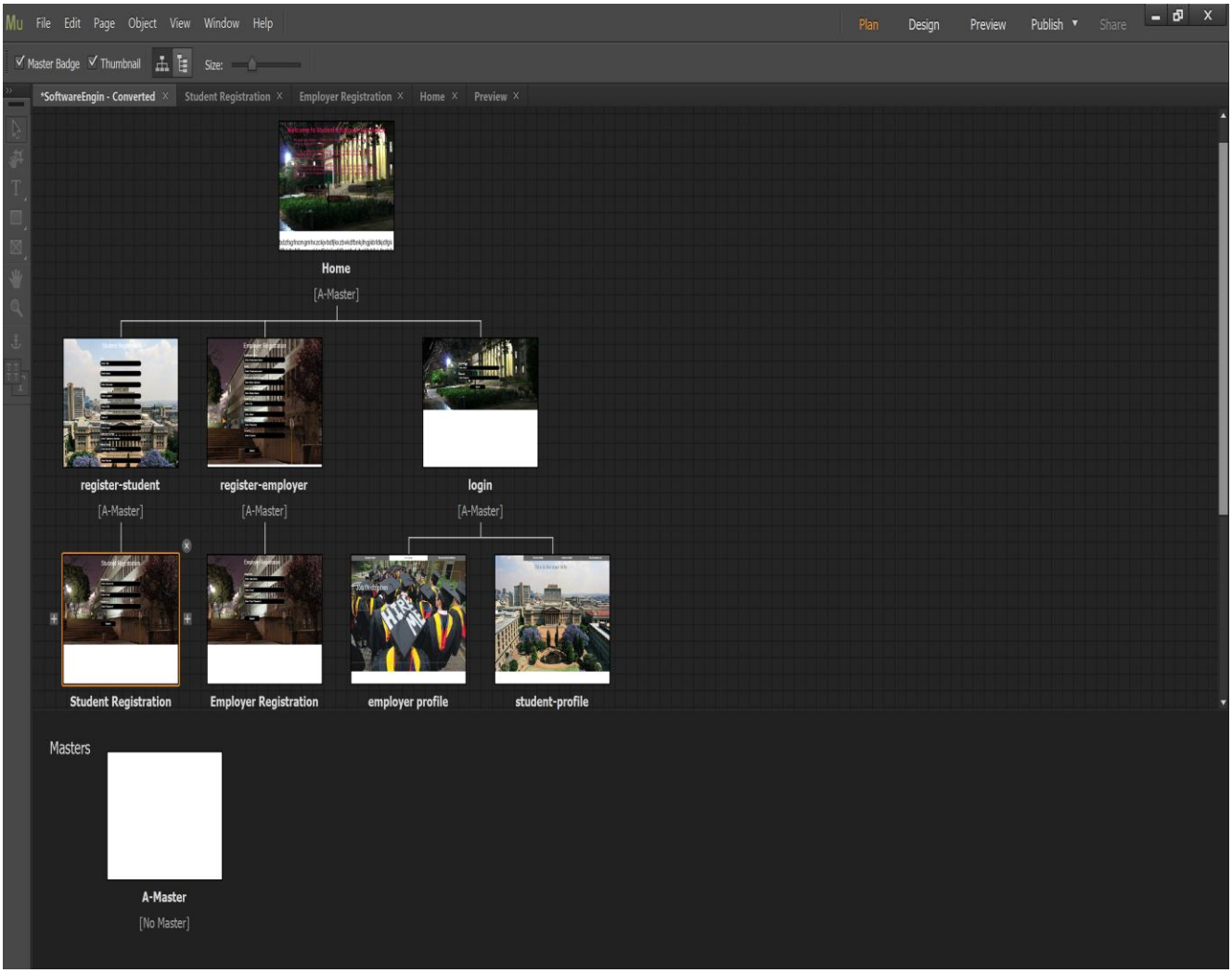
Backend

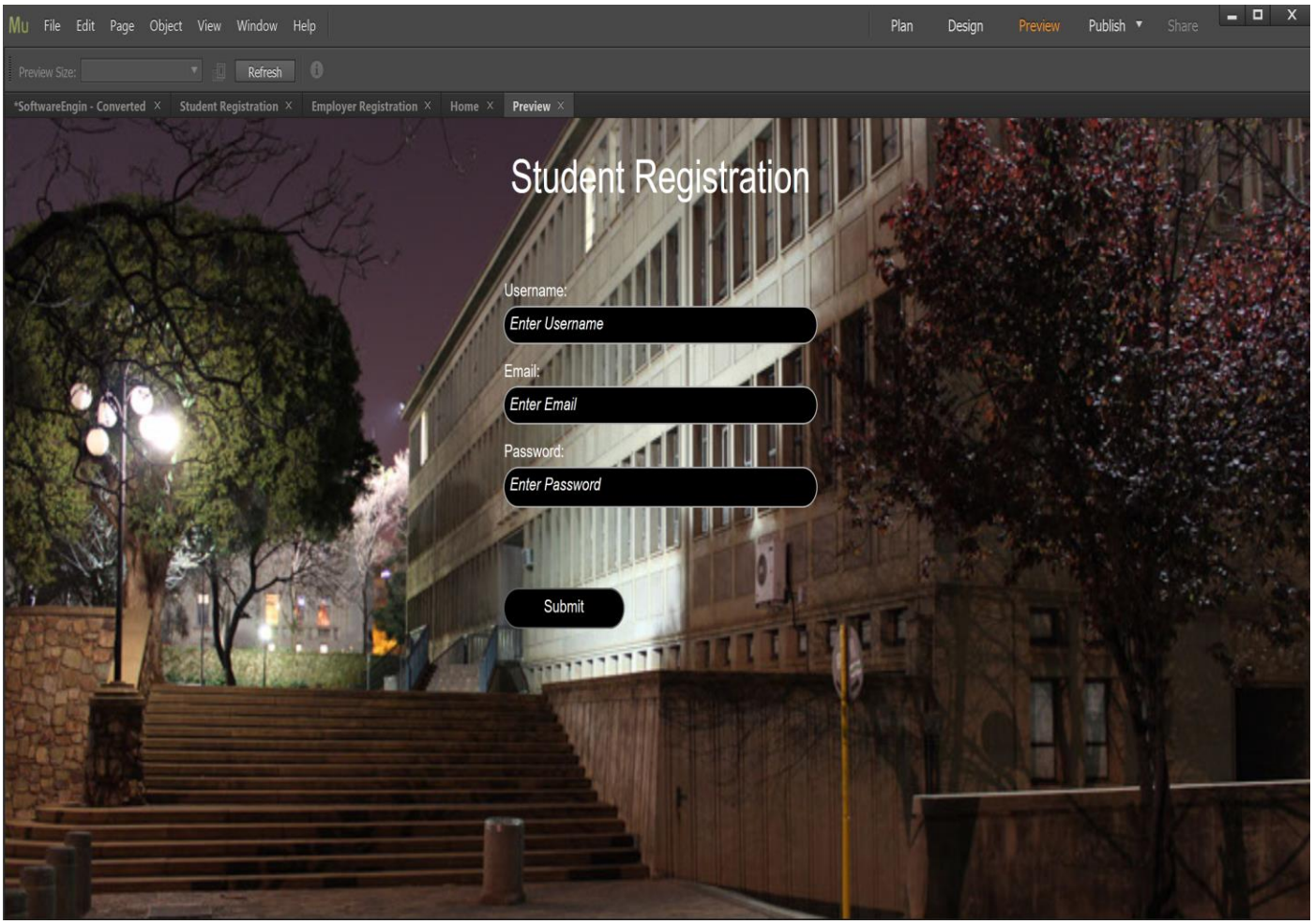
- Complete employer database
- Php for login
- Php for register

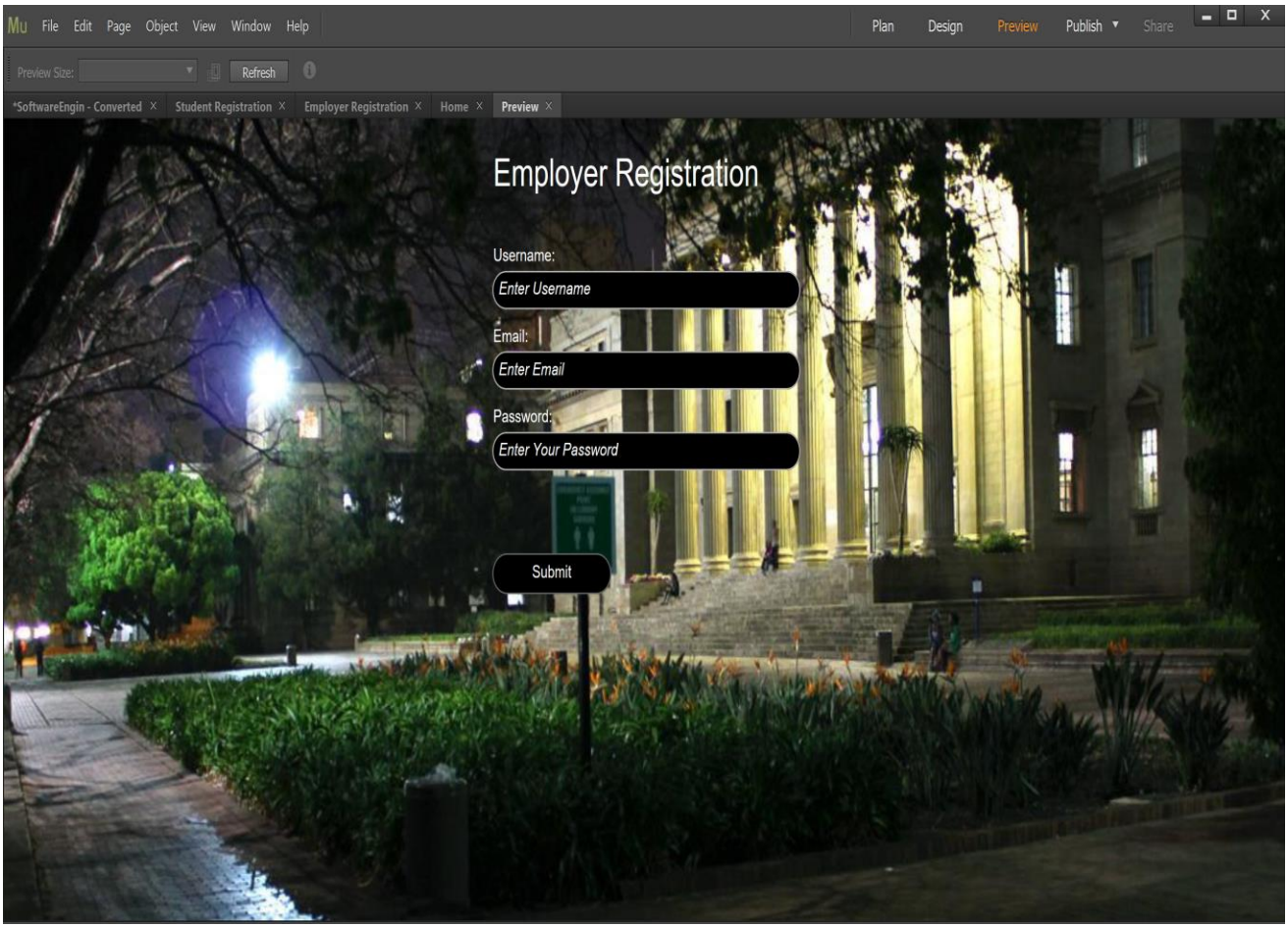
Frontend

- Complete forms
- Interface as the pages
    1. Student home
    2. Employer home

5.3 Sprint 1 Retrospective




Mu
File
Edit
Page
Object
View
Window
Help
Plan
Design
Preview
Publish
Share
Preview Size:
Refresh
*SoftwareEngin - Converted
Student Registration
Employer Registration
Home
Preview
Student Registration
Username:
Enter Username
Email:
Enter Email
Password:
Enter Password
Submit


Mu File Edit Page Object View Window Help
Plan Design Preview Publish Share
Preview Size:
Refresh
*SoftwareEngin - Converted
Student Registration
Employer Registration
Home
Preview
Employer Registration
Username:
Enter Username
Email:
Enter Email
Password:
Enter Your Password
Submit

Meeting Minutes

What went wrong in the last sprint?

- We discovered a few limitations and problems on Adobe Muse
- We had encountered problems with submitting forms
- We did not like the prototype we had on muse
- It lead to database errors which caused us to redo the database

What is the way forward?

- We are going to use Mobirise
- Mobirise has more advantages
- Mobirise has more possibilities
- And we are going to use both XAMP and PhpStorm