Assignment 1

Task 1.1: Build a group

DAFOT Group

Member	Contact info
Dev Sharma	dev.sharma@rwth-aachen.de
Anton Okolnychyi	anton.okolnychyi@rwth-aachen.de
Frederik Zwilling	frederik.zwilling@rwth-aachen.de
Oleksandr Zastupailo	oleksandr.zastupailo@rwth-aachen.de
Tatiana Vert	tatiana.vert@rwth-aachen.de

Wiki page: https://www3.elearning.rwth-aachen.de/ws14/14ws-14118/collaboration/Lists/WikiList1/Group%20DAFOT.aspx

Task 1.2: First concept of your project

a) Develop an idea for a new and innovative website that can support learning.

The idea of our project is aimed to create highly scalable and extensible web service for people who teach any programming course with exercises. Attendees should download the template of exercise, implement own solution and upload it to our web service. After that these exercises can be verified by our service automatically (by running unit tests which were implemented by the course owner). Results of unit tests will be available both for attenders and organizers.

System will let organizers provide description to each test explaining purpose and common mistakes of it. In addition, attendee will be able to see a list of exceptions in failed tests. All of these can be used as a hint to improve and reimplement solution for the exercise.

The web service should maintain diverse languages and should be developed in a way that allows to add new programming languages in convenient and easy manner.

Our system will consist of separated modules, which in future could be embedded into a comprehensive learning website (even L2P).

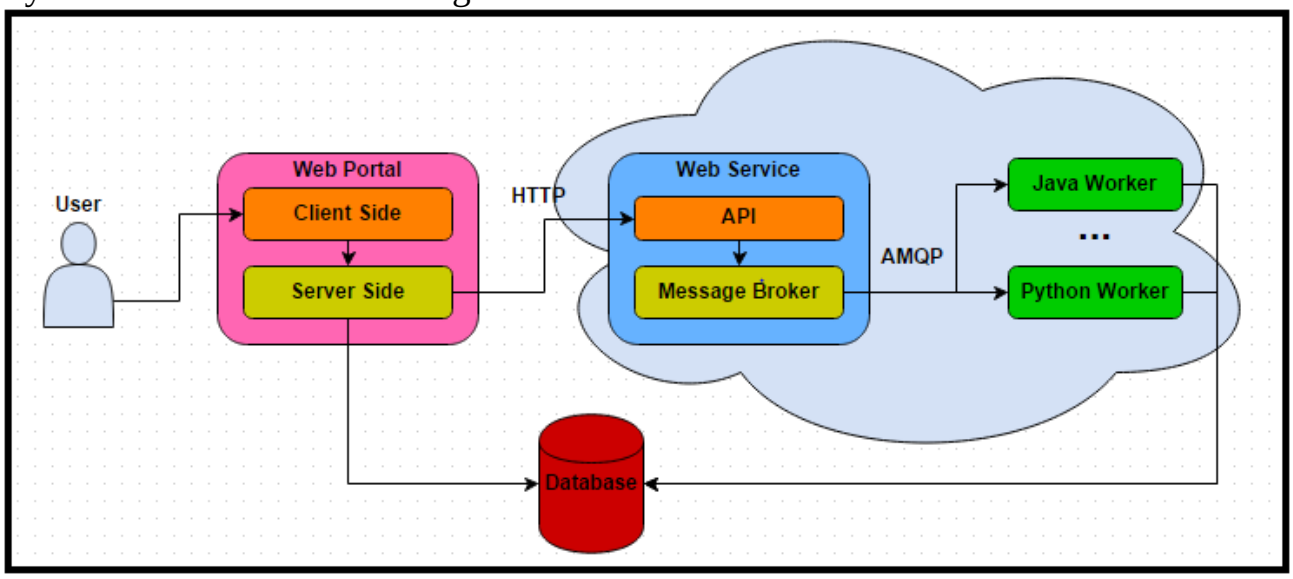
b) Collect and structure features that your website should support. Give examples and/or descriptions for every feature.

Main features:

- highly scalable;
- easy to extend;
- compilation check for particular languages (e.g. Java);
- unit tests execution.

System Architecture

System will have the following architecture:



Web Portal is a web application, which is responsible for interaction with users, managing users and sending requests for code execution to Web Service via HTTP.

Web Service is a core of the system. Its main task is to forward requests to appropriate worker. Communication between Web Service and Worker will be done via AMQP.

Worker is a verifier of attendee's solutions.

Web Portal can be replaced by any learning web site (e.g. L2P), which requires the functionality of automatic code verification. Moreover, each programming language will be executed on a separate worker, which can be deployed in a cloud. This makes our service **highly scalable**.

To support any additional programming language should be provided one more worker instance with the implemented logic of code verification. The key benefit of this architecture is the ability to use already existing Web Service API. For this reason, our application is **easy to extend**.

User Types

System distinguishes two types of users. The following table describes the use cases for each type of users.

Attendee	Organizer
• creates a profile; • sees available classes; • asks for the invitation for the class; • navigates through the exercises list; • chooses exercise, downloads the template and uploads code solution through web form or via file;	• creates profile; • creates a class; • defines available programming languages for solutions; • adds set of the exercises;

• views the results of compilation, unit tests execution; • sees the feedback from Organizer with message for particular test.	• specifies unit tests for each exercise; • invites attendees to class; • approves request to class; • have history of all attempts per each attendee.

c) Perform a market study and search for similar websites and compare their features with your concept

Resource	Description	Benefits	Disadvantages
	**Udacity** is a MOOC that uses a very similar environment to provide implementation exercises. But the value of the feedback on the solution is very limited. Programming classes use the Python language; programming assignments are graded by automated grading programs on the Udacity servers. Example of a feedback for an incorrect solution: “Incorrect. You didn't pass the first test case listed.” Nevertheless, there is some very basic feedback such as compile errors and “Your code didn't display any output”.	• possibility to make a code review of the code assignment attempt.	• not open-sourced solution, organizer should create a course on Udacity to have the possibility to check code assignments; • supports only Python graders, other languages require to install additional environment
	**EdX** is a massive open online course provider and online learning platform. EdX differs from other MOOC platforms, such as Coursera and Udacity, in that it is nonprofit and runs on an open-source software platform. EdX architecture is divided into modules which can be used separately on any learning platform. One of these modules is **edX XServer**. **XServer** accepts student code submissions and runs the code using courseware graders. It uses AppArmor to control execution of code. AppArmor is a Linux kernel security module, which allows the system administrator to associate with each program a security profile that restricts the capabilities of that program.	• open-sourced solution; • XServer can be used separately from other edX modules.; • big community.	• can’t be deployed in a cloud; • supports only Python graders from the box; • does not support course exercises on multiple languages; • there are a lot of requirements and dependencies to install.

coursera	**Coursera** is a well-known platform for massive open online courses. Coursera is powered by Amazon Web Services. It uses Amazon S3 to store data, that provides high scalability, reliability, high speed and low-cost storage infrastructure. Coursera allows students to solve quizzes and submit their code for programming assignments. Submitted code is automatically graded by a set of tests. They allow to estimate correctness, memory usage and timing of program. The report usually consist of summary and details. Tests report either passed or failed.	• works with Amazon Web Services, so is executed in the cloud; • supports different languages (e.g. Java, Python, C).	• not open-sourced solution, organizer should create a course on Coursera to have the possibility to check code assignments;

Code execution examples

edX:

```
12      for month in range(12):
13          month += 1
14          tempBalance -= lowestPayment
15          tempBalance += (annualInterestRate/12) * tempBalance
16      if tempBalance < 0.1 and tempBalance > -0.1:
17          break;
18      if tempBalance > 0:
```

Correct

Test results

See full output

CORRECT

See full output

Hide output

INCORRECT

Test: edge dictionary

Your output:

```
The variable `edges` should contain a dictionary
Got instead variable of type <type 'str'>
```

Correct output:

```
Examining edges...
True

Test completed
```

Hide output

Hide output

INCORRECT

Test Case 1

balance = 320000; annualInterestRate = 0.2

Your output:

```
Lowest Payment: 29157.51


*** ERROR: Your numerical answer was not within the error margin of our answer.
First error found: expected 29157.09 +/- 0.2, got 29157.51. ***
```

Correct output:

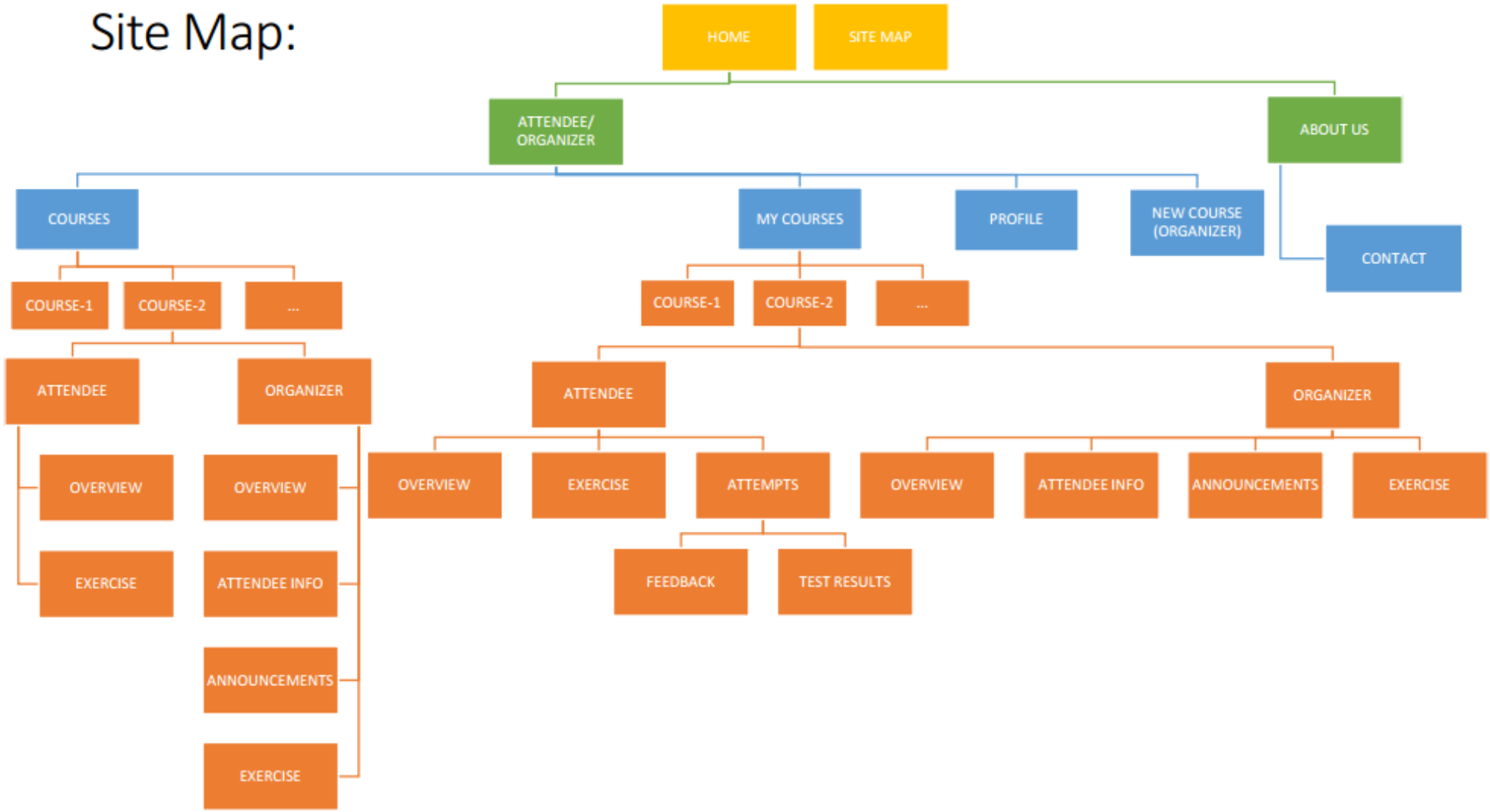
```
Lowest Payment: 29157.09
```

Coursera:

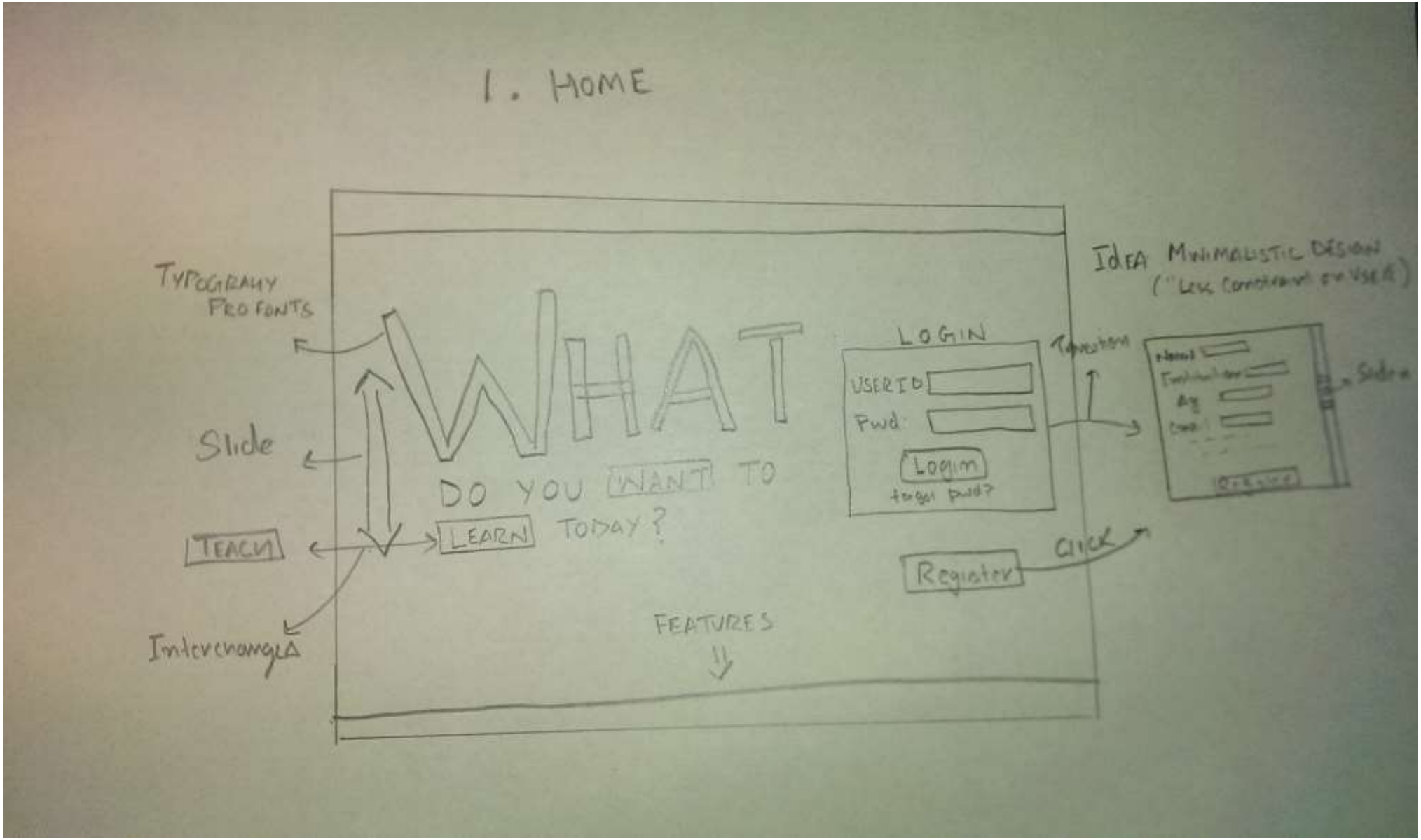
```
Compilation:  PASSED
Style:        FAILED
Findbugs:     No potential bugs found.
API:          PASSED

Correctness:  35/35 tests passed
Memory:       39/51 tests passed
Timing:       24/24 tests passed

Aggregate score: 97.65% [Correctness: 65%, Memory: 10%, Timing: 25%, Style: 0%]
```

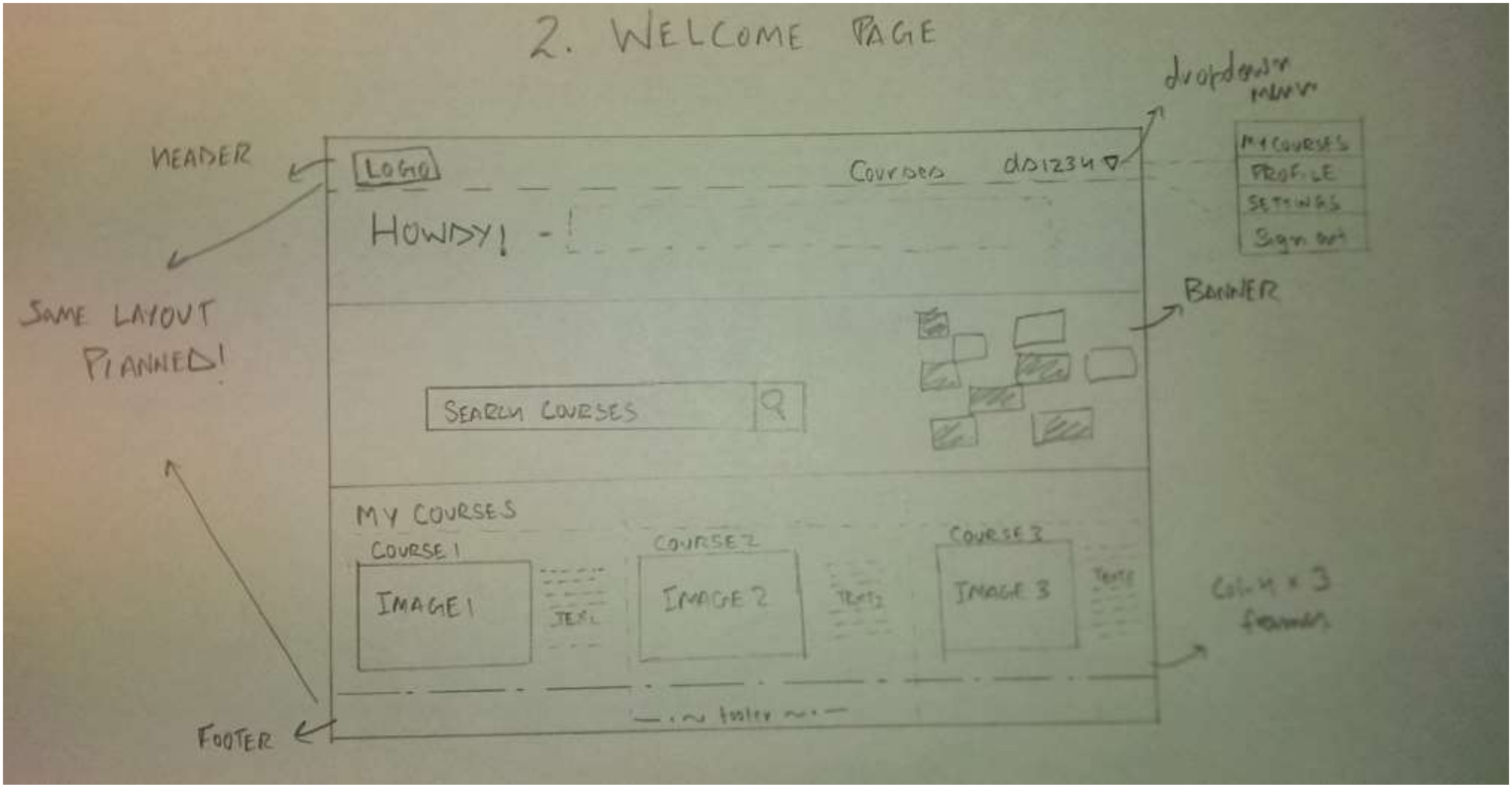
d) Draw a site map that shows the overall organization of your website.

Site Map:

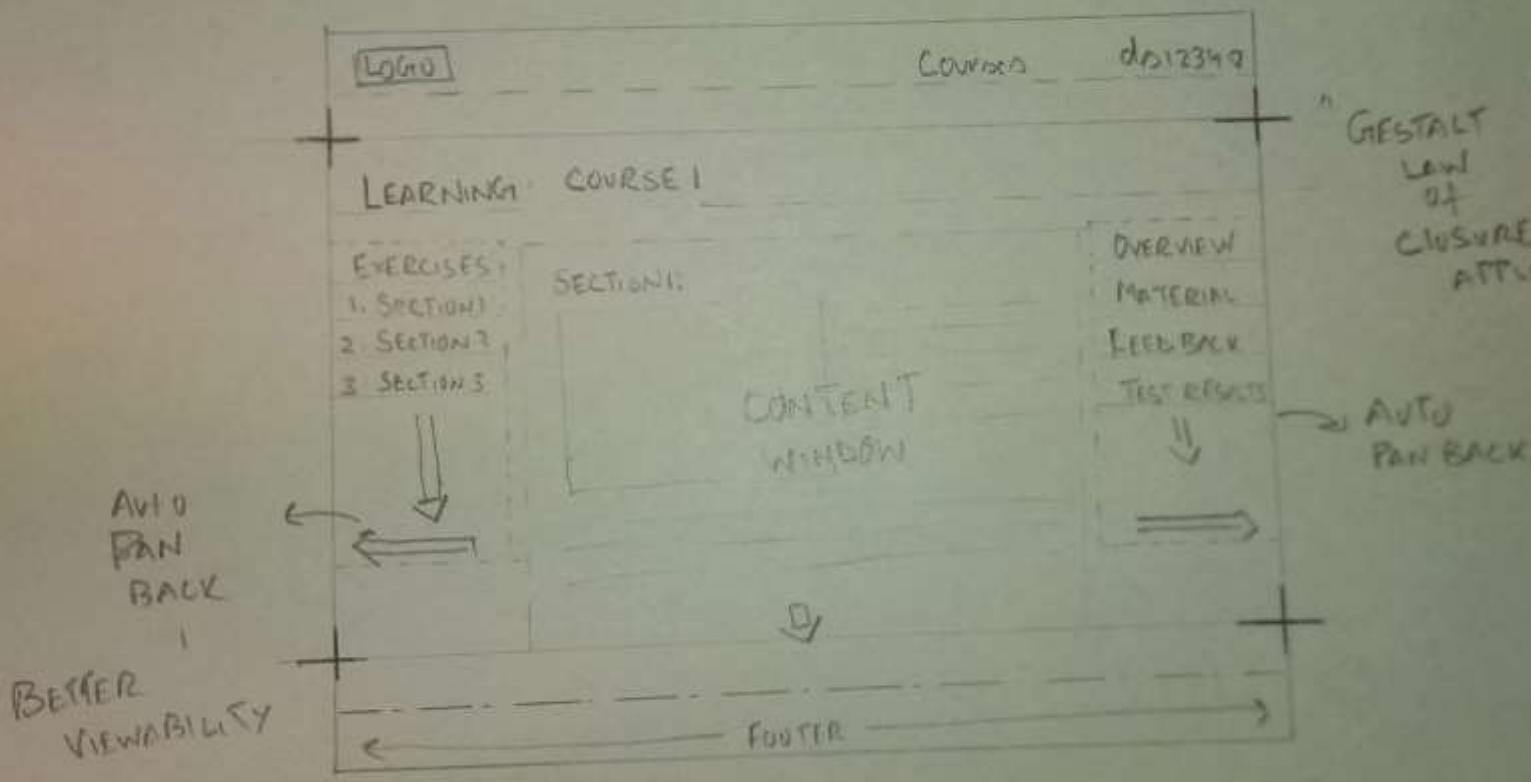

HOME
SITE MAP
ATTENDEE/ ORGANIZER
ABOUT US
COURSES
MY COURSES
PROFILE
NEW COURSE (ORGANIZER)
CONTACT
COURSE-1
COURSE-2
...
COURSE-1
COURSE-2
...
ATTENDEE
ORGANIZER
ATTENDEE
ORGANIZER
OVERVIEW
EXERCISE
OVERVIEW
ATTENDEE INFO
ANNOUNCEMENTS
EXERCISE
OVERVIEW
EXERCISE
ATTEMPTS
FEEDBACK
TEST RESULTS
OVERVIEW
ATTENDEE INFO
ANNOUNCEMENTS
EXERCISE

e) Draw and describe a first layout draft for your website.




2. WELCOME PAGE
HEADER
LOGO
COURSES
dd1234
dropdown menu
MY COURSES
PROFILE
SETTINGS
Sign out
SAME LAYOUT PLANNED!
HOWDY! -
BANNER
SEARCH COURSES
MY COURSES
COURSE 1
IMAGE 1
COURSE 2
IMAGE 2
COURSE 3
IMAGE 3
Col-4 x 3 format
FOOTER

3. COURSE PAGE



Task 1.3: Student talk topic

a) Choose 3 different technologies, you would like to present in a 30 minutes (max) group presentation and describe each one briefly.

RabbitMQ is open source message broker software (sometimes called message-oriented middleware) that implements the Advanced Message Queuing Protocol (AMQP). The RabbitMQ server is written in the Erlang programming language and is built on the Open Telecom Platform framework for clustering and failover. Client libraries to interface with the broker are available for all major programming languages.

Django is a free and open source web application framework, written in Python, which follows the model–view–controller architectural pattern. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself.

MariaDB is a drop-in replacement for MySQL. MariaDB strives to be the logical choice for database professionals looking for a robust, scalable, and reliable SQL server. To accomplish this, the MariaDB Foundation work closely and cooperatively with the larger community of users and developers in the true spirit of Free and open source software, and release software in a manner that balances predictability with reliability.

b) Rank your ideas from 1 to 3 so that we can see what you would like most when we assign the topics.

1. RabbitMQ;

2. Django;

3. MariaDB.