

# **Experimentation system – Requirement Document**

**Gilad Chen 207168568, Gili Chiko 209296326,**  
**Sean Naor 206911315, Shachar Morad 206781080**

## **Abstract**

In our project, we will develop a system for experiments in the field of software engineering for the Web environment. This system will enable the creation of experiments, participating in them, grading them etc. The experimentation system's purpose is to enable the comparison between different programming languages and their uses by different groups of programmers.

## **Contents**

<b><u>Chapter 1 - Introduction</u></b>	<b><u>4</u></b>
<u>1.1 The Problem Domain</u>	<u>4</u>
<u>1.2 Context</u>	<u>4</u>
<u>1.3 Vision</u>	<u>5</u>
<u>1.4 Stakeholders</u>	<u>5</u>
<u>1.5 Software Context</u>	<u>5</u>
<b><u>Chapter 2 - Usage Scenarios</u></b>	<b><u>6</u></b>
<u>2.1 User Profiles - The Actors</u>	<u>6</u>
<u>2.2 Use-cases</u>	<u>6</u>
<u>2.2.1 User stories</u>	<u>6</u>
<u>2.3 Special Usage Considerations</u>	<u>7</u>
<b><u>Chapter 3 - Functional Requirements</u></b>	<b><u>8</u></b>
<b><u>Chapter 4 - Non Functional Requirements</u></b>	<b><u>12</u></b>
<u>4.1 Implementation Constraints</u>	<u>12</u>
<u>4.2 Platform Constraints</u>	<u>12</u>
<u>4.2.1 SE Project Constraints</u>	<u>13</u>
<u>4.3 Special Restrictions &amp; Limitations</u>	<u>13</u>
<b><u>Chapter 5 - Risk Assessment and Proof Of Concept</u></b>	<b><u>14</u></b>
<b><u>Appendices</u></b>	<b><u>15</u></b>
<u>Cost Analysis</u>	<u>15</u>
<u>Glossary</u>	<u>15</u>

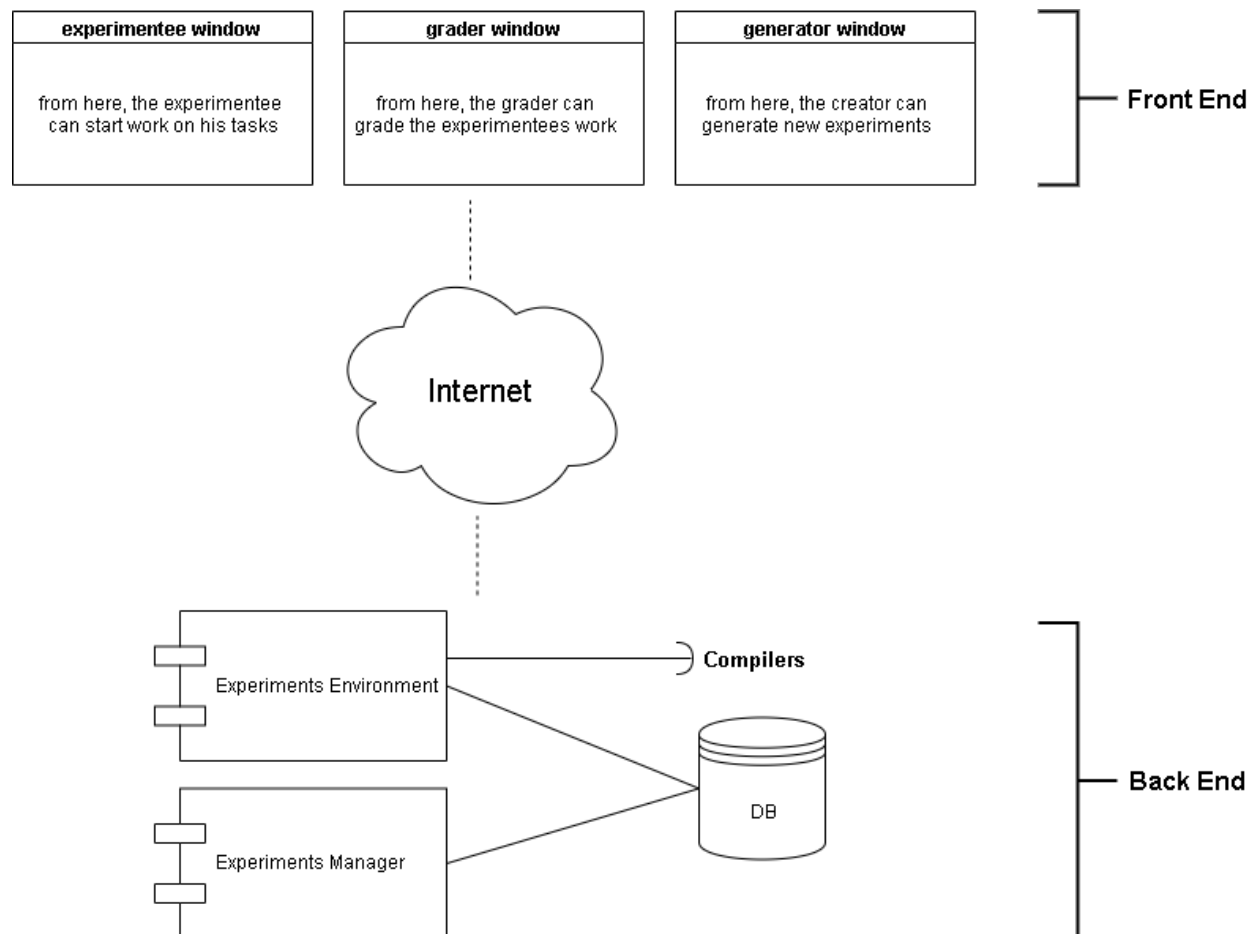
# Chapter 1 - Introduction

## 1.1 The Problem Domain

There are many programming languages with various paradigms of programming and various purposes. The usability of languages in different applications can be tested by different kinds of parameters - number of lines, generality of different parts of the code and so on.

## 1.2 Context

Our system will have two main modules, one for generating experiments and one for participating in an experiment. A part of participating in an experiment requires running, checking and compiling different code scripts, therefore we will use external compilers for testing different coding languages.



## 1.3 Vision

The broad vision of the system is to enhance the ability of software developers and researchers to determine and quantify certain attributes of programming languages while comparing them in different coding scenarios.

## 1.4 Stakeholders

Software researchers, experimentees, outsourced graders, programming language developers

## 1.5 Software Context

The experiment management will allow creation of new experiments with different stages. In a scenario in which a user would like to create new experiments, he will input the different desired stages of said experiment.

The experiment environment will present users with a programming experiment in which they will have to follow a set of instructions defined in the experiment management. Once a step is completed it is unchangeable.

# Chapter 2 - Usage Scenarios

## 2.1 User Profiles - The Actors

**Experiments creators** - User of the experiment management module. An endpoint user with an idea for an experiment with relevant connection to some kind of programming languages. The experiment creator will supply the needed data of the experiment and use the experiment management module to create it.

**Experimentees** - User of the experiment environment module. An endpoint user with knowledge in the field of programming. The experimentee will supply the answers and solutions, to his understanding, as required by the experiment.

**Graders** - User of the experiment system. An endpoint user with knowledge and experience in the field of programming. The grader will analyze and grade the answers given by the experimentees.

## 2.2 Use-cases

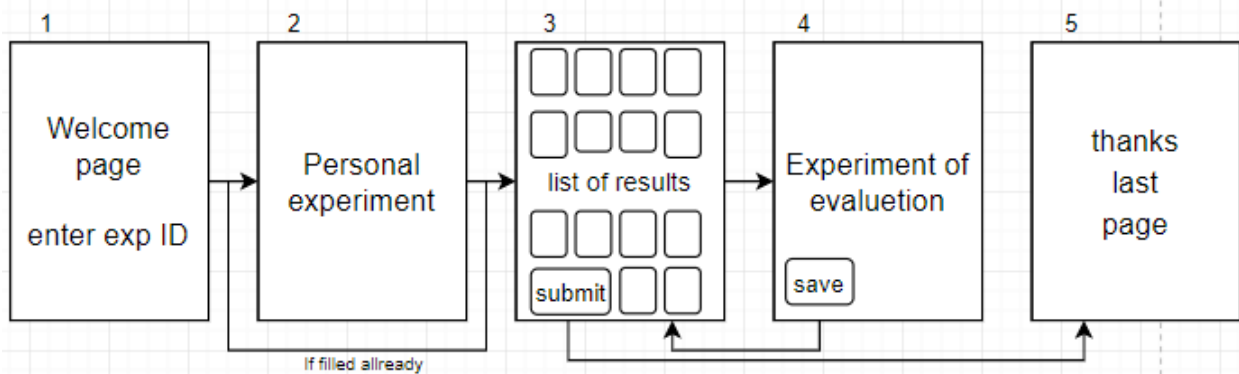
**Creating an experiment** - The user will login to the system as an experiment creator, will ask to create a new experiment, follow the system instructions and supply data to his choice.

**Participating in an experiment** - The user will open an experiment, follow the system instructions and supply data to his choice in order to complete the experiment.

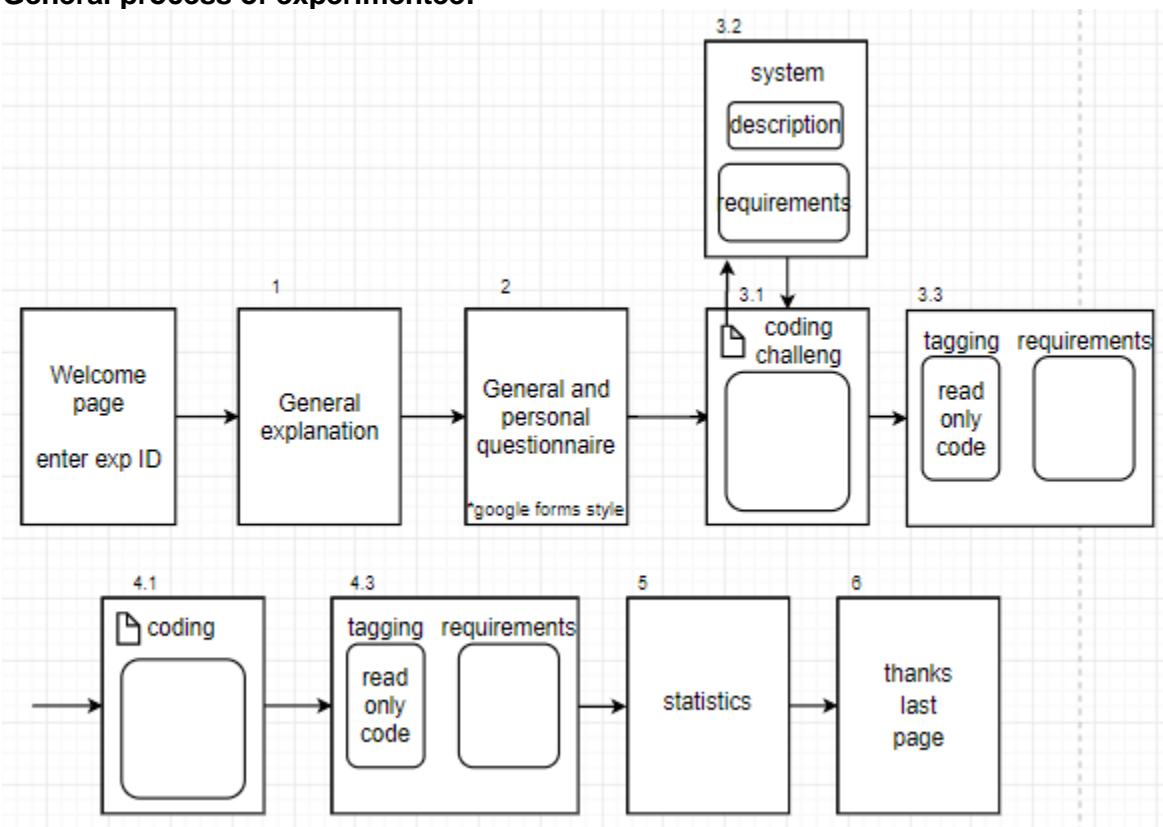
**Grading an experiment** - The user will login to the system as a grader, the system will present him the experiments results that he has permission to grade, the grader will choose a certain experimentee results and supply data to his choice in order to grade the answers.

## 2.2.1 User stories

**general process of grader:**



**General process of experimentee:**



**general process of experiment creation:**

A creator should enter the system and ask to create a new experiment. An experiment should have a basic form (like the general one above), and could contain more pages as wish. An experiment could also contains grading tasks, that is a personal experiment for each grader, and experiment for each experimentee results, which contains option to displays the experimentee results (the creator would choose which stages to display to the grader). The creator should enter the details necessary for any stage, and when finished he can view the all experiment and get two codes/ids, one for experimentees and the other for graders.

## 2.3 Special Usage Considerations

Web development kits for different programming languages will be needed in order to use certain programming languages in each experiment.

## Chapter 3 - Functional Requirements

**P - Priority:** \_\_\_\_\_ **R - Risk:** \_\_\_\_\_

**MH - Must have** \_\_\_\_\_ **H - High**

**SH - Should have** \_\_\_\_\_ **M - Medium**

**NH - Nice to have** \_\_\_\_\_ **L - Low**

#	Requirement	P	R
1	Creator	-	-
1.1	The system will allow a user to sign up and log in using the BGU username and password.	MH	L
1.1.1	Some BGU users will be listed in the system as admins and are the only ones allowed in the experiment management system.	MH	L
1.1.2	Admin users can manage admins - add or remove users from the admin list.	SH	L
1.2	The system will allow the logged in user to create a new experiment.	MH	H
1.2.1	The user will create and fill in the stages from given stage formats.	MH	M

1.2.2	The stages can appear in what order the creator decides.	<b>MH</b>	<b>L</b>
1.2.3	The user will create grading tasks	<b>MH</b>	<b>L</b>
1.2.3.1	The user will bind graders to each grading task	<b>MH</b>	<b>M</b>
1.2.3.2	The user will bind experimentees to each grader	<b>MH</b>	<b>M</b>
1.2.3.3	The user will create personal experiment for the graders to answer once	<b>MH</b>	<b>M</b>
1.2.3.4	The user will create evaluation experiment for the graders to fill in for each result	<b>MH</b>	<b>M</b>
1.2.4	The system will allow the logged in user to copy one of his other experiments and edit them, thus creating a new experiment from an old experiment's template.	<b>NH</b>	<b>H</b>
1.3	By selecting an experiment, the system will allow the following operations	-	-
1.3.1	View the experiment's stages	<b>SH</b>	<b>L</b>
1.3.2	Generate access codes for experimentees or graders	<b>MH</b>	<b>M</b>
1.3.3	View experimentees' results	<b>MH</b>	<b>M</b>
1.3.4	View graders' results per grading task	<b>MH</b>	<b>M</b>

1.3.5	Make the experiment inactive	SH	L
1.3.6	Update the experiment only if no experimentee participated.	NH	M
1.3.7	Can export the results of the experiment or of a given experimentee to a csv \ JSON file. Might be more than one file	NH	H
1.3.8	Can export the definition of an experiment to a XML / JSON file	NH	H
2	<b>Grader</b>	-	-
2.1	The grader can enter his/her access code to see specific grading task and the results that have been binded to him/her	MH	M
2.1.1	The grader should fill in the evaluation experiment for each experimentee's result	MH	L
2.2	The grader can see which results were already graded.	SH	L
2.3.1	The grader can fill in an evaluation experiment about an experimentee without submitting it	NH	M
2.3.2	The grader can edit all evaluation experiments that yet to be submitted	NH	M
2.3.3	The grader can submit all the evaluation experiments, after submitting there is no option to edit them	NH	M

2.4	The grader must fill in his personal experiment before submitting any evaluation experiment	SH	L
3	<b>Experimentee</b>	-	-
3.1	An experimentee can start his/her participation in an experiment by entering a given code.	MH	L
3.2	The experimentee will perform the given tasks in each stage (if it has any tasks) and will advance to the next one.	MH	H
3.3	Once moving a stage, the experimentee cannot go back.	MH	M
3.4	The system will save the user's progress every time he/she advances a stage.	MH	H
3.5	The user will have an option to manually save his/her progress in the coding stage.	NH	L
4	<b>Experiment</b>	-	-
4.1	An experiment <b>must have</b> a programming task	MH	H
4.2	An experiment <b>must have</b> a tagging task that will follow the programming task	MH	H
4.3	An experiment <b>must have</b> an information page	MH	L



4.4	An experiment <b>must have</b> a questionnaire page	MH	M
4.5	An experiment <b>must have</b> a statistics page	MH	H
4.6	The system will have a real time developing environment within the programming tasks of an experiment that will:	-	-
4.6.1	Allow coding	MH	M
4.6.2	Allow running code	SH	H
4.6.3	Allow basic debugging	NH	H
5	<b>Allies (Management Users with permissions in a specific experiment)</b>		
5.1	Allies can access only experiments they have permissions to	NH	M
5.2	Allies can only perform actions on experiment if they have permission to	NH	M

## Chapter 4 - Non Functional Requirements

#	Requirement	P	R
	The system should be generic - will allow to create all sorts of experiments with the stage's formats.	MH	M

2.	The system should allow multiple experimentees to participate in an experiment simultaneously.	MH	M
3.	The system should allow multiple experimentees to perform different experiments in parallel.	SH	L
4.	Every stage of the experiment will be recorded in a database.	MH	H
5.	Every once in a while, automatically saves the experimentee progress	NH	L
6.	The web pages will be given in HTML5 format	MH	L
7.	The websites will be designed to be supported by Google Chrome and Mozilla Firefox browsers.	MH	L

## 4.1 Implementation Constraints

**Safety & Security** - The data from an experimentee will not be accessible to other experimentees and will be accessible to graders and the experiment creator.

**Portability** - The two modules are web services and so, will be accessible from the Google Chrome browser.

**Usability** - All users will have at least basic programming knowledge and ability.

## 4.2 Platform Constraints

The project is a Web project and so, it needs to be developed in a web development environment.

### 4.2.1 SE Project Constraints

We will need a minimal amount of experimentees to test performing an experiment with the system.

We will need a server connected to the internet to allow us to test the entire system.

## 4.3 Special Restrictions & Limitations

There are no special restrictions and limitations to this system.

# Chapter 5 - Risk Assessment and Proof Of Concept

For our proof of concept, we will design the following experiment:

1. We will set up an online database with web API.
2. We will set up a web page with a menu and some general information.
3. The web page will lead to a module of coding with some outsourced web platform for coding in a specific programming language.
4. Then we will have a tagging task about the coding segment.
5. We will give the option to add a different programming language in the coding segment.
6. Then we will show how we can get the results from the database after some experiments have submitted their results for the experiment.

## Appendices

### Cost Analysis

- The proof of concept experiment will involve volunteering students and will have no cost.
- Every outsourced development kit and other code snippets will be open sourced.

### Glossary

- Experiment - A custom made task with a number of stages. This task is defined for the purpose of testing specific criterias and hypotheses.
- Grading task - Part of an experiment for the graders use. Would contain a personal experiment for the grader to fill once, and another experiment based on the experimentees results.
- Personal experiment - a series of stages, belong to a grading task, such that the graders should fill in once.
- Evaluation experiment - a series of stages, belong to a grading task, such that the graders should fill in for each experimentees' results.
- Experiment management - A module in the project for designing and creating experiments.
- Experiment environment - A module in the project for performing a designed experiment.
- Development kit - A tool for Writing code, running and debugging a specific programming language.
- Software researcher - A person with knowledge in the field of software that wants to examine a theory or an idea.

- Stage formats - Formats for creating a stage that are: information page, questionnaire, code writing, code tagging, statistics page.
- Information page - Will contain textual data given by the researcher.
- Questionnaire - Will contain input components such as: radio button, text box, check box, toggle etc.
- Code writing page - Will contain an online embedded development environment for code writing that can run code with an option to display the given requirements.
- Code tagging - A split screen with the requirements on one side and the previously written code on the other.
- Ally - A management user that gets permissions to a specific experiment.