Ayala Bouhnik-Gelbord
Eden Dahary
Yosef Dassa

system for checking algorithms
Software Design Document

Date: (12/12/2022)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

This SDD document describes the architecture and system of the components of the website that computerized system for checking algorithms and the precise implementation details required to satisfy the requirements as specified in the PRD document.

It is assumed that the reader has read the PRD, since this document also defines the implementation details of the desired behavior given the requirements within it.

Also, this document provides a description of the design of the system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

1.2 Scope

Intended for computer science students who want to practice algorithm problems, this is a networking web site that allows students to study, test themselves and track their progress. Unlike traditional web sites such as HackerRank, our product allows lecturers to monitor students' grades and progress.

Projected benefits for the consumer - the student receives feedback when he submits a solution to the programming algorithm.

Projected benefits for the lecturer – the lecturer receive the student's score when the student submits a solution.

Our goals are to *improve student skills* in efficient way:
1. to give the student test case that allow the student learn from them.
2. to provide feedback on the student code.
3. to allow the lecturer to monitor the student progress

1.3 Overview

-System overview- (page: 3)

-System Architecture-architectural design, decomposition description and And design rationale (page: 3-6)

-Data design- data description and dictionary (page 6-7)

-Component design- (page 7-8)

-Human interface design- overview, screen images and screen objects and actions (page 9-10)

Reference Material

- Vision Statement
- PRD Document

2. SYSTEM OVERVIEW

Our website is built on an open-source framework. Source code is available using standard open-source management tools such as Git. All source code is stored in a GitHub repository(https://github.com/AylaBouhnik). Everyone can contribute to the website. The website was written mainly in Python. GitHub is used as the software development platform.

In summary, the system design includes the following sub-systems:

- Algorithm Website
  - Test Cases
  - Algorithms
- PyCharm Development Platform
- GitHub Version and Source Code Control

There is no expectation that any of these systems will be changed or modified with the proposed system.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

Login Page

Used as a main page if the user is not connected the page includes:

- Login as a guest - Enter your name
- Login button
- Login button using email and password

If the input successfully passed the tests, the login button will execute

Server request for user authentication.

Once verified, the user will be taken to the main page and saved

The cookie we received for server-side authentication on requests

Future.

In failure, you will receive an error message as received from the server.

The error message will be displayed in the dialog window.

Links

- Home page link to main page

- Run algorithm
- Save code
- Start new task
- Change profile
- Logout will delete the cookies and return the user to main page
- Login

You can see below in illustration 1-the description of the log in.

website structure

- Main page:
  - Login
  - Sign up
  - Sign up as a lecturer

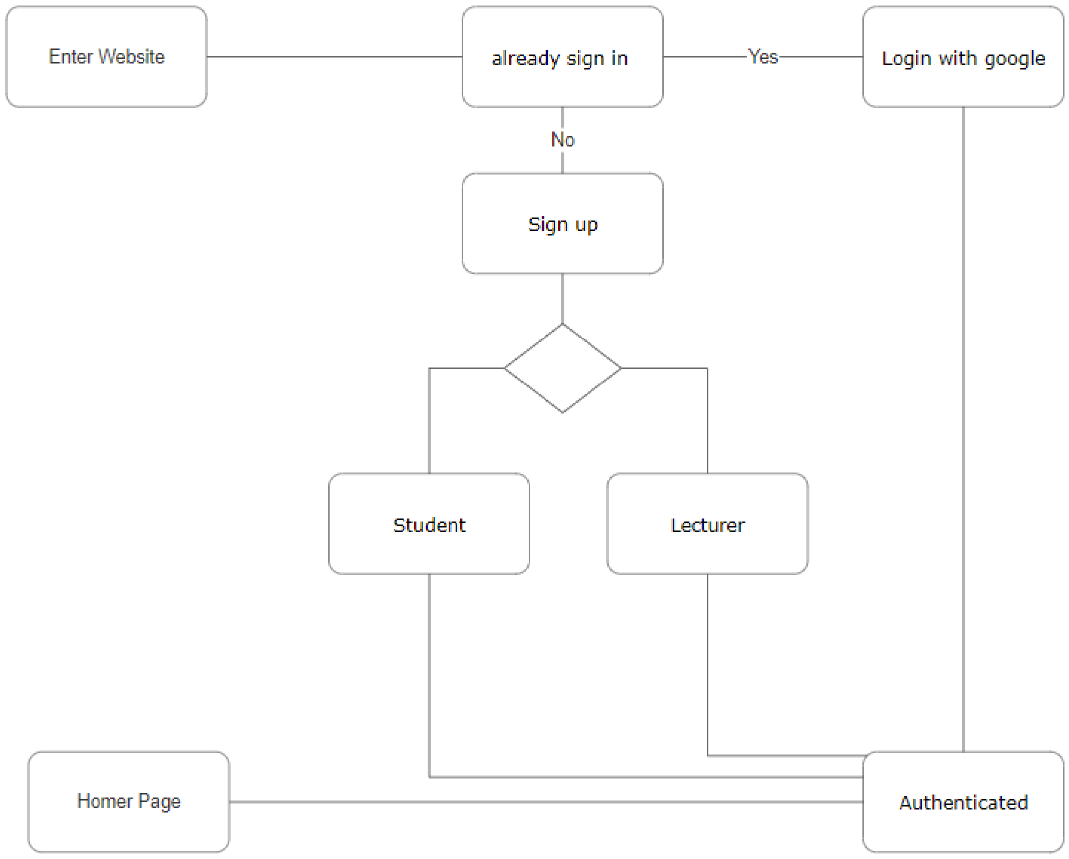
- Home page:
  - Search algorithm
  - Practice
  - Start test
  - Profile
  - My grade and feedback (from lecturer)
  - Exams dates

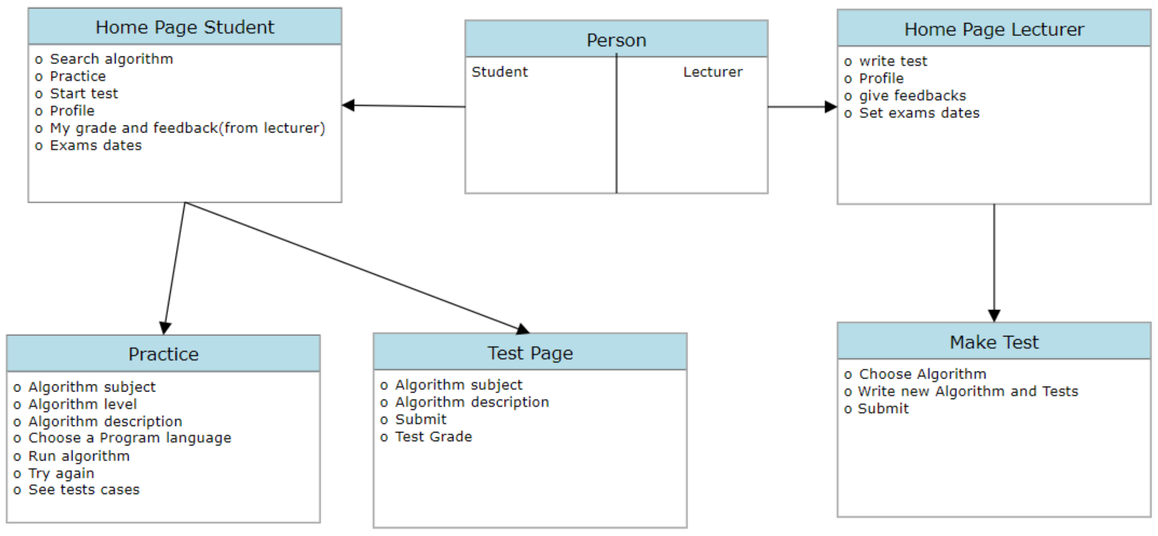
- Practice page:
  - Algorithm subject
  - Algorithm level
  - Algorithm description
  - Choose a Program language
  - Run algorithm
  - Try again
  - See tests cases

- Test page:
  - Algorithm subject
  - Algorithm description
  - Submit
  - Test Grade

- Account types:
  - Lecture – can upload tests and give feedback to the students.
  - Student - can practice, see the exams dates, answer the lecturer tests, and read feedback.

3.2 You can see below in illustration 2 the description of the objects of the game structure (in 3.2). Decomposition Description





3.3 Design Rationale

Person is a super entity.

Lecturer and Student are sub-entities inherited from Person.

Algorithm template contains all the algorithms.

Test template contains all the test cases for each algorithm.

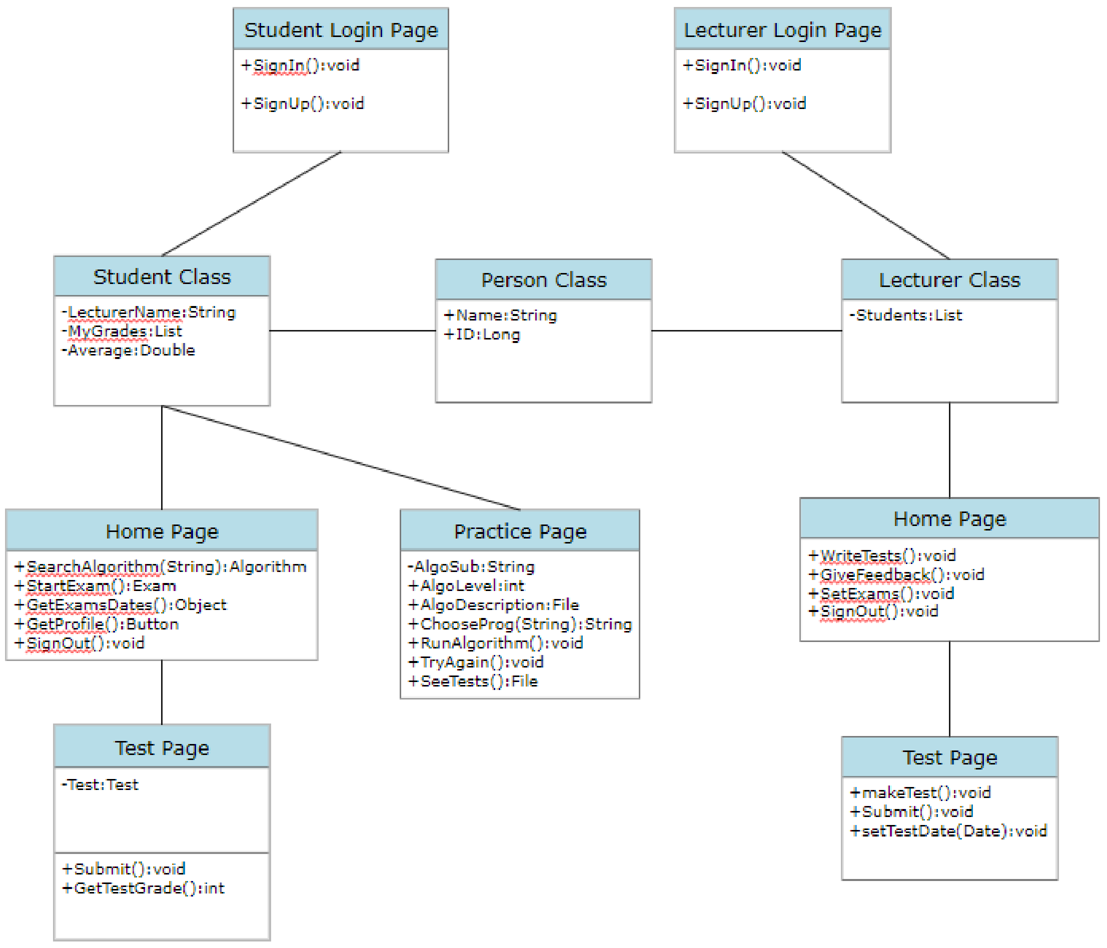
Database that will save all the users, grades etc.

4. Data Design

4.1 Data Description

With firebase any registered user in google can be connected by his google account and login his details are saved by using firebase automatically. The user profile picture, grades, feedbacks etc. will be saved in firebase storage.

4.2 Data Dictionary



5. Component Design

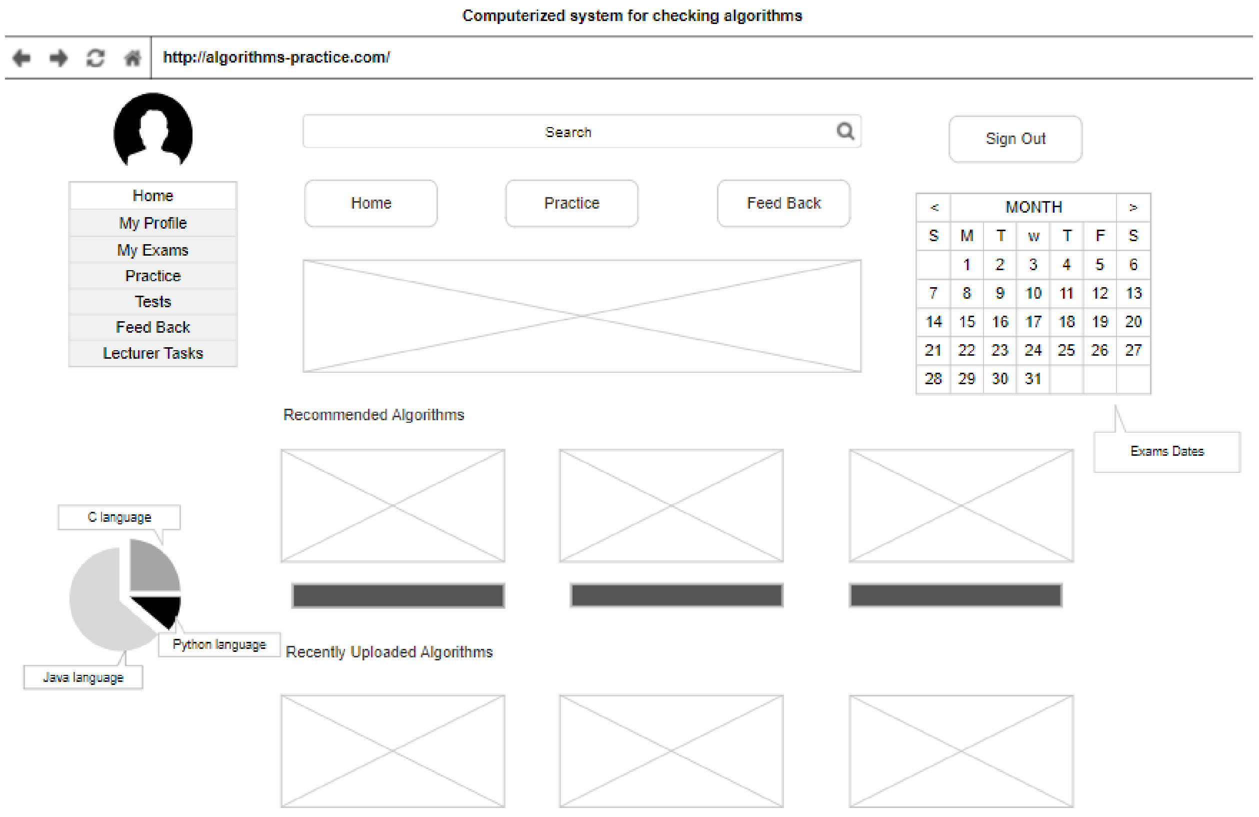
```
SignIn()-
Get email and password:
    If Verified - Open Home Page
    Else – Try again
SignUp()-
Get all the requested details:
    If Not Null – Create New Account
```

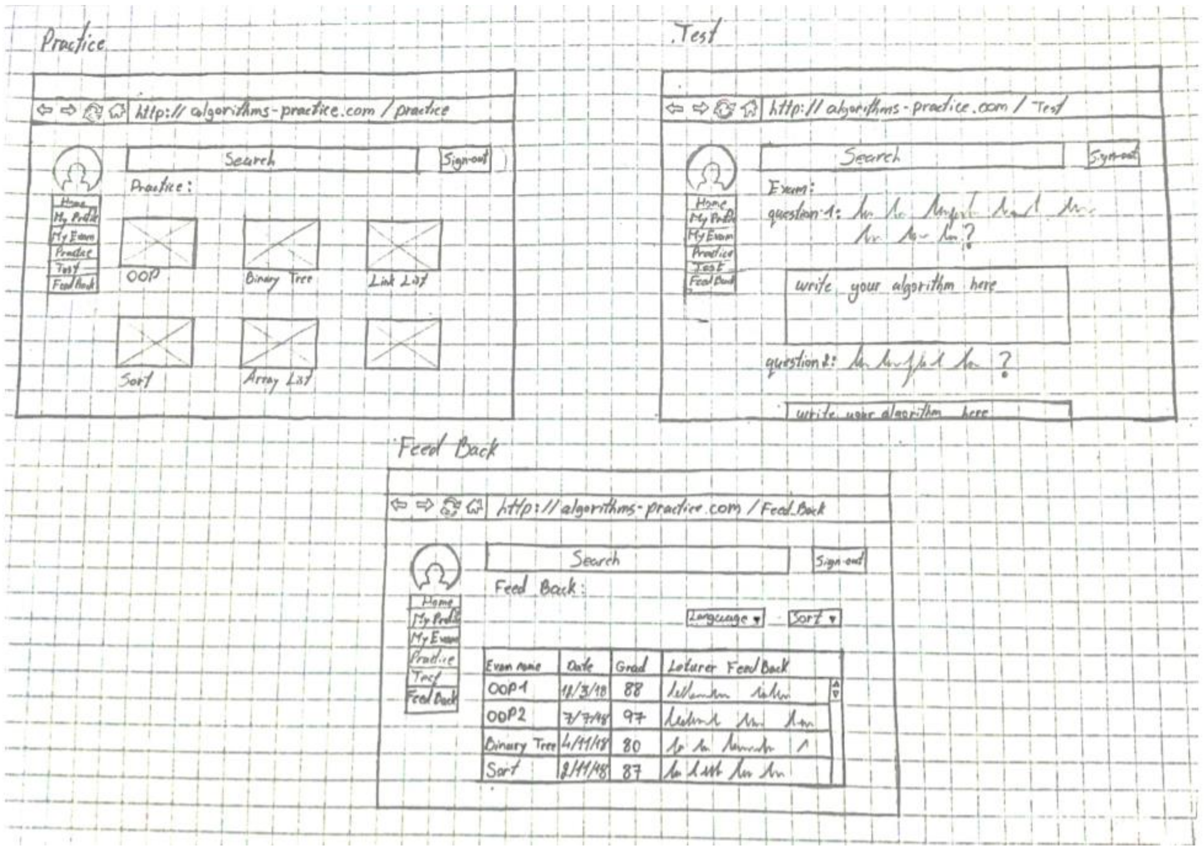
```
        Else – Try again
    SignOut()-
        If Button Clicked – Sign Out

SearchAlgorithm(String algo)-
        If algo exists if our database – openAlgo()
        Else "this algorithm does not exist"
StartExam()-
        If Button Clicked – OpenExam()
GetExamsDates()-
        Return all the exams dates that the lecturer set
ChooseProg(String lang)-
        If lang equal java open java
        If lang equal python open python
        If lang equal C++ open C++
RunAlgorithm()-
        Run the Current Algorithm
TryAgain()-
        Give another opportunity to solve the challenge
SeeTests()-
        Return all the test cases
WriteTest()-
        Give the lecturer an opportunity to write a test for his students
GiveFeedback()-
        Give the lecturer an opportunity to write feedback for his students
Submit()-
        Give the student/lecturer to submit the test
GetTestGrade()-
        Return the Grade
```

6. Human Interface Design

6.1 Overview of User Interface

Our project is a website, where you can practice algorithm and do tests.





Exam name	Date	Grad	Leturer FeedBack
OOP1	18/3/18	88	[illegible]
OOP2	7/7/18	97	[illegible]
Binary Tree	4/11/18	80	[illegible]
Sort	8/11/18	87	[illegible]

Screen Images

My Exams- history of all the exams (code, grades, feedback).

Practice - history of all the practices.

Test – contain exam that the lecturer upload.

Lecturer Task – contain all the task that the lecture chose.

Feedback – contain the lecturer feedback on the tasks and exams of each student.

6.2 Screen Objects and Actions

it relies on the drawing in the previously paragraph.

- Search algorithm – allows you to search a specific algorithm to practice.
- Practice – all the algorithms you can practice on.
- Start test – answer the test that the lecturer provides.
- Profile – contain your picture profile, ID, milestones etc.
- My grade and feedback (from lecturer) – contain grades and feedback.
- Exams dates – contain all your exams dates (rely on the dates that the lecturer provide).