Cairo University
Faculty of Engineering
Computer Engineering
Ahmed S. Kaseb

CMPN203
Software Engineering
Final Exam
Spring 2020

Deadline: 6 June 2020 at 11:59:59 p.m.	Total Marks: 40

Applied Project: Building A Streaming Service Using Modern Technologies

The purpose of the project is to design and implement a software product using the state-of-the-art tools and technologies in the software industry. The required product should mimic some functionalities of the popular streaming service: Spotify [Web API].

Product Features

The following are some examples for the features that can be implemented.

- User Management Module
  - Login
  - Signup and mail confirmation
  - Authorization for protected routes
  - Facebook login
  - Reset password using email
  - Edit password
  - Edit profile data
  - Types of users (free, premium, artist)
  - Convert free users to premium users using email confirmation
- Notifications & Sharing Module
  - Push notifications
  - Recent activities to show notification history
  - Share songs
  - Notifications should be sent for the following events: following an artist, a followed artist has added a new song or album, or another user has liked a generated playlist.
- Artist Module

    - Add, edit and delete albums
    - Add, edit and delete songs
    - Show statistics for uploaded songs and albums. Statistics should include: number of listeners, number of likes, and charts for the number of listeners and likes per day, month, and year.
- Audio Player
    - Play, pause, stop, next, previous, playing queue and volume control
    - For android, a widget is required.
    - Like a song and following an artist
    - Show songs by genres
    - Search for songs
- Playlist Manager
    - Add (and remove) songs to a playlist
    - Play playlist sequentially
- General Module
    - Home page and static pages for help (at least 5 pages)
    - Player home page similar to open.spotify.com

Team-Specific Requirements

Each team (2 to 5 members) must select one of the following areas:

- DevOps
- Backend
- Frontend
- Android
- Testing

Each team member must develop at least 4 mid-level features from the features listed above. Other comparable features can be selected as well. **Each team must select features that were not previously implemented by the same team during the coursework project.**

Each team has specific requirements:

1. DevOps Team
    - Use the cloud or any other hosting service with HaaS.
    - Manage Linux-based systems: logs, remote access, automation, scripting.
    - Set up tools for monitoring and automation (e.g., Jenkins or Travis-CI).
    - Very quick deployments.
    - Auto scaling on the cluster level and the services level [Optional]
    - Use Kubernetes [Optional]

    - Dockerization [Optional]
2. Backend Team
    - Construct a complete requests-collection with an API Client (a requests sender) to allow sending all types of requests with different major parameters. [Example tool: Postman].
    - Unit testing with coverage above 95%
    - Auto-generate documentation for the REST APIs and the source code.
    - Authentication and authorization for applicable API requests.
    - Design and implement a valid database system. The concepts of migrations and seeds should be used to quickly initiate and populate the database for easy and incremental deployment.
    - Use Docker [Optional]
3. Frontend or Android Team
    - Unit testing with coverage above 95%
    - Responsive [will be tested on browsers with various screen sizes].
    - Make a service to mimic the backend responses (with fake data) to be able to work without the backend whenever needed. Alternating between the real and the mimic backend should be as smooth as possible with global configuration modifications only.
    - Use AJAX.
4. Testing Team
    - End-to-end testing (E2E):
        - For web: 90% coverage
        - For Android: 90% coverage
    - Stress testing

Deadline & Submission Guidelines

- The deadline of delivering the project is **6 June 2020 at 11:59:59 p.m.**
- You must submit a single compressed file containing all the following items:
    - A document with the full names and IDs of all team members.
    - A fully-integrated working software.
    - Link to the source code Git repository. The TA must also have access to the repo.
    - Final source code files.
    - Documentation. Depends on the team as detailed above.
    - Demo video showing the product and all its functionalities. Time limit: 5 minutes.
    - Individual progress videos to explain the contributions of each team member. Each member must record a separate video. Time limit: 2 minutes per member.
    - A Document for:

    - Distribution of the final tasks (i.e., who did which features)
    - Challenges the team faced (both technical and non-technical)
- The compressed file must be submitted using Google Classroom by each and every student. This means that a team of five members will submit the same exact compressed file (including all the materials above, even including the progress videos of all the teammates) five times (one submission from each student).

Evaluation Criteria

DevOps

1. Professionalism [10%]
2. Scripts for tools installation (on new server) [25%]
    a. Scripts [70]
    b. Understanding (High Level) [25]
    c. Code Style [5]
3. Deploy required software and libraries on actual server [20%]
4. Integration [30%]
    a. Run on fresh repo (SSH, scripts, and so on). [25]
    b. Testing [20]:
        i. Run E2E tests [5]
        ii. Run Unit tests [5]
        iii. Deploy on Success [5]
        iv. Email on failure [5]
    c. Build frontend, backend and integrate them [30]
    d. Email sending for the backend [5]
5. Automating the process (Git Hook) + Continuous Integration (Frontend + Backend)[15%]

Backend

1. Professionalism [10%]
2. Functionality [45%]
    a. Functions [100]
    b. Validation [-10]
    c. Authentication [-10]
    d. Authorization [-10]
    e. Restful & Stateless [-10]
    f. Modularity & Single Responsibility [-20]

    g. Configuration places [-5%]
3. Code Style [10%]
4. Unit Testing [10%]
5. Functional Documentation [5%]
6. Integration [10%]
7. DB Design [5%]
8. API Documentation [5%]

Frontend or Android

1. Professionalism [10%]
2. Functionality [35%]
    a. Functions [100]
    b. Validation [-10]
    c. Authentication [-10]
    d. Authorization [-10]
    e. Quality of Frontend (Professionalism) [Good UX, UI] [-10]
    f. Restful & Stateless [-10]
    g. Modularity & Single Responsibility [-20]:
        i. Files & modules: 80
        ii. Separated CSS, HTML, JS (with tolerance): 20.
3. Mock Services [5%]
4. Code Style [10%]
5. Unit Testing [10%]
6. Functional Documentation [5%]
7. Integration [10%]
8. Responsive Screen [laptop Size, tablet Size, Mobile Size] [10%]
9. Responsivity [Don’t halt until the server responds] (Loading…) [5%]

Testing

1. Professionalism [Factor of 10%]
2. E2E [80%]
    a. Unreported Ordinary Bug [-5]
    b. Unreported Indirect corner-case Bug [-2.5]
3. Stress Test [10%]
    a. Multiple Types of requests. [10]
    b. Meaningful Report [60]
    c. Report Understanding [30]

General Rules

- Pagination [Partial Retrieval] is a must whenever applicable.
- All testing should cover direct and corner cases. Testing effects of actions should be thorough. Naive assertions won't be accepted neither in unit-testing nor in E2E testing.
- Cheating will result in zero grade, and your project will be rejected.
- Integration is required (and graded) within each team.
- Huge commits are forbidden.
- Pushes that don't work or don't comply with the practices chosen may be penalized.
- There will be neither a discussion nor an oral exam, so make sure you present your work thoroughly. The submitted file must contain everything.
- Push the code periodically (on time). No major pushes allowed.
- Documentation:
    - All the used tools.
    - License allowance of your product, for example, if you use a tool that prohibits commercial use, you should indicate that your product cannot be commercial.
    - Code should be acknowledged whenever used from an external resource (website, open-source code, tutorial and so on). Any copied code without clear acknowledgement - inside the source code, in the readme.md and at the final documentation- will be considered cheating.
- Students are assumed to belong to the same teams and areas as the classwork project unless they request otherwise. The deadline for changing your team is 14 May.

Support

If you have any questions, please feel free to communicate with one of the instructors at any time using any of the following methods:

Name	Email	Google Meet (by appointment)
Ahmed Kaseb	akaseb@eng.cu.edu.eg	https://meet.google.com/dxm-gybx-uux
Youssef Ghatas	youssef.s.ghatas@gmail.com	https://meet.google.com/uwb-sgcf-zrm
Mahmoud Khaled	mahmoud.k.a@outlook.com	https://meet.google.com/zsv-zctz-acm
Amira Amer	amira.a.amer@gmail.com	https://meet.google.com/grh-ciqs-riy

The subject of all emails should have the following template: "[SWE2020] <Subject>".