Bilkent University

Department of Computer Engineering

Senior Design Project

3KET: 3D Event Venue Viewing Application

Project Specifications Report

Ege Kaan Gürkan, Musa Ege Ünalan, Mehmet Çalışkan, Doğa Tansel, Mustafa Göktan Güdükbay

Supervisor: Asst. Prof. Dr. Eray Tüzün

Jury Members: Asst. Prof. Dr. Shervin Arashloo and Asst. Prof. Dr. Hamdi Dibeklioglu

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1. Introduction

Nowadays, people use technology to buy or reserve tickets for events. There are websites or applications like Ticketmaster, Biletix where users buy tickets for the events they want to attend. In general, these websites present only textual information or 2D visualization on the seats and tickets. People who buy tickets without actually knowing the view from the seats may have problems during the event. We believe that having a 3D perspective view showing how the scene appears from a particular seat will help customers choose their seats informed. **Figure 1** displays the Berlin Philharmonie Seating Plan to show how different seats are positioned according to the stage.

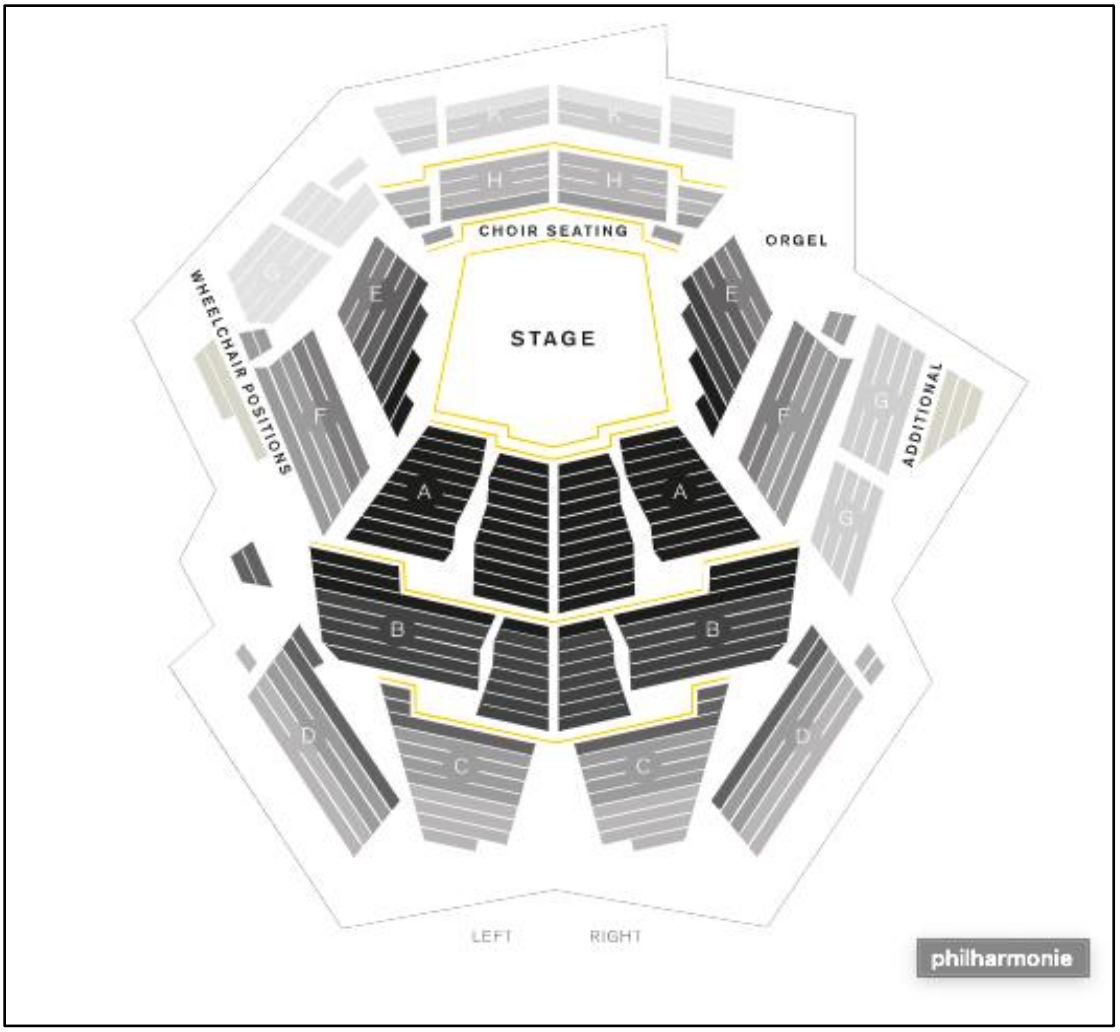

Figure 1. Berlin Philharmonie Seating Plan [1].

1.1. Description

In our application, businesses will upload three-dimensional (3D) models of venues such as concert halls and stadiums. **Figure 2** shows an example 3D model, the 3D model of the Carnegie Main Hall. Using the 3D model, users will be able to look from different viewpoints by selecting seats and see how the stage is viewed from the selected seats. In this way, the users will decide on their seat. Businesses will match the seats in the 3D model with tickets and they can show ticket information using our application. If the users decide to buy a ticket for a seat that they select using our application, our application will redirect them to websites entered by event creators where businesses usually sell tickets, such as Biletix and Ticketmaster, to continue their payment.

The business owners can load a 3D model they built on another platform and match the seats with tickets using an editor that we will provide. After matching the tickets with seats, the 3D models will be uploaded to an S3 bucket. The customers will load the 3D model using a browser application, and they will be able to roam freely in it and choose a seat to purchase. The users will be able to walk through the 3D model and look at the perspective of different seats. The customers will be able to write comments on the viewpoints of the seats they buy and the venue. This way, businesses will be able to get feedback and improve their platform and 3D models. Our application will make event recommendations to users using their previous events and their specified hobbies.

We will also extract events from ticket distribution websites to display them to the customers. Furthermore, the customers will be able to upload photos, and videos, to share their memories of an event they visited. Users can view the images and videos uploaded by other users. There will be two types of leaderboards. In the first leaderboard, the events passed and got the highest interaction in the past

day, week, month, and year. In the second type, there will be the events that will happen in a week and get the highest interaction.

Figure 2. Carnegie Main Hall 3D Model [2]

1.2. Constraints

1.2.1. Implementation Constraints

- 3D models will be stored in Amazon Simple Storage Service (Amazon S3) [3].

- Unity will be used to develop the editor for matching seats with tickets [4]. Then, the editor will be exported as a WebGL program [5],[6].

- We will store data related to the users in Amazon Relational Database Service (RDS) [7].

- We will use Amazon Web Services Lambda (AWS Lambda) in our application to run our backend services [8].

- We plan to use Python to manage the backend services. We will use the Django framework to develop the server part of our application [9].

- We plan to use React to design our user interface [10].

- We plan to use GitHub [11] and Git Version Control System [12] for managing the project code.

1.2.2. Economical Constraints

- Finding or producing 3D models can be costly. We plan to use the free models we find on Unity Asset Store [13], 3D warehouse [14] or further and develop on them. We also plan to create 3D models for some venues in Bilkent by ourselves using prebuilt objects, stages, floors on Blender [15] and Unity [4]

- Generally, Amazon services cost money. However, we plan to use AWS Free Tier [16].

- Amazon S3 [3] storage that offers 5GB storage is free for 12 months.

- Amazon Lambda [8] is always free until 1 million requests per month.

- Amazon RDS [7] offers 750 hours per month for MySQL, PostgreSQL, MariaDB, Oracle BYOL, or SQL Server is free for 12 months.

1.2.3. Technological Constraints

- Developing 3D models for test purposes on Unity and Blender can have difficulties.

- Selecting the seats in the 3D model from an external editor can cause problems.

- Low-tier devices can have issues while loading the 3D models.

- To avoid issues related to the configuration of React in the development stage, we plan to use Docker Containers for the front-end development [17].

1.2.4. Sustainability Constraints

- The technology of 3D models can change in the future. We plan to develop an application where users in the future will not experience any compatibility issues with the created models.

- We are planning to support various 3D model formats like OBJ, FBX, GLTF to increase the compatibility of our application.

1.3. Professional and Ethical Issues

First of all, our application will store user information only to perform basic functionalities such as logging in and making recommendations. Our application will not share user information with other applications. Additionally, the users will be able to report inappropriate content in photos and videos shared by other people. Lastly, the users will be able to comment on the views of seats and how they match the actual venue. Their names will be kept confidential and will not be visible to the venue owners.

2. Requirements

2.1. Functional Requirements

2.1.1. Business Client Functionalities

- Businesses should be able to create, edit and delete events, add descriptions to them and assign websites for redirection during payment.

- Businesses should be able to upload the 3D models they designed using different applications such as but not limited to Blender, Sketchup, Unity, and WebGL. The application should be able to accept the formats OBJ, FBX, and glTF.

- Businesses should be able to match each seat in the 3D venue model with a ticket (assigned to that seat). Furthermore, presented with a birds-eye view, our application will automate the seat marking process after matching some seats by calculating the positions of the yet unmatched seats (e.g., by labeling the seats of a row between the beginning and end seats of that row).

- Businesses should be able to see the reviews written by users about their events or their venues.

- Businesses should be able to add animations of events on the stage (animation of players playing soccer or an orchestra playing music) to make the view realistic and attractive.

2.1.2. Customer Functionalities

- Users should be able to create a profile by entering their information, selecting their preferences and hobbies.

- Users should be able to see the perspective for each seat that is matched by the venue owner and roam inside the 3D venue.

- Users should be able to see events in a categorized manner and search particular events according to location, genre, artist, or venue.

- Users should be able to see the recommended events by the application according to their past events, recent searches, preferences, and hobbies.

- Users should be able to see ticket information for the seats they chose in the 3D model.

- Users should be able to share photos, videos, and reviews about the events they participate in.

- Users should be able to look at the event leaderboards, containing events that passed and got the highest interaction in the past day, week, month, and year.

- Users should be able to look at the rank of events depending on their interaction that will happen at most in a week’s time.

2.2. Non-functional Requirements

2.2.1. Usability

- Businesses should be able to match seats with tickets in the editor only relying on the user manual.

- Customers should be able to look at the perspectives of different seats and proceed to ticket checkout websites by selecting the seat they wish to book only relying on the user manual.

- Customers should be able to search, view, and favorite/mark events only relying on the user manual.

- Customers should be able to upload videos, photos, and reviews for events they have attended by only relying on the user manual.

2.2.2. Reliability

- Businesses should be able to upload their 3D models using our service to the server without encountering any technical problems. They should also be able to edit or add some features to the models of their venues.

- Customers should be able to view 3D models for the venues and proceed to ticket checkouts on other websites using our browser application without encountering any technical problems.

2.2.3. Extensibility

- Our application can be extended to add new functionalities and interactive features for users and features on different types or genres of events and 3D modeling.

2.2.4. Maintainability

- The application can be modified in order to accept new 3D model formats that can be produced in the future.

2.2.5. Accessibility

- The users should be able to view the 3D models from popular web browsers like Google Chrome, Mozilla Firefox, Opera, Safari.

2.2.6. Scalability

- The application will be able to automatically scale the resources to support the incoming requests without any additional configuration [18].

3. Glossary

Glossary for any domain-specific terms that we use in our report.

Amazon Simple Storage Service (Amazon S3): It is a service that presents scalability, data availability, security, and performance for storing data [3].

Amazon Relational Database Service (RDS): It is a service that proves relational database functionalities in the cloud [7].

Amazon Web Services Lambda (AWS Lambda): It is a serverless compute service that runs code without managing servers, generating workload-aware cluster scaling logic, sustaining event integrations, or controlling runtimes [8].

Unity: It is a cross-platform game engine [4].

WebGL: It is a cross-platform, web standard for a 3D graphics API based on OpenGL ES (Embedded Systems) used with HTML5 Canvas element supported by browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera [5].

Docker: Docker is a platform that uses operating system virtualization to develop software in packages called containers [17].

OBJ: It is a file format used in 3D printing and 3D graphics applications [19].

FBX: It is a file format used to edit 3D geometry and animation [20].

glTF: glTF is a standard file format for loading 3D models in game engines and applications developed by Khronos Group [21].

React: It is a Javascript library for producing user interfaces on the web [10].

Django: Django is a high-level Python web framework that can be used for both backend and frontend development [9].

4. References

[1] Philharmoniker, B. (n.d.). *Berliner Philharmoniker*. Aboseries - Philharmonie | Berliner Philharmoniker. Retrieved October 10, 2021, from https://www.berliner-philharmoniker.de/en/concerts/floor-plan-price-list-aboseries-philharmonie/.

[2] TurboSquid (by ShutterStock). (2020, October 10). Carnegie Main Hall 3D model. TurboSquid - 3D Models. Retrieved October 10, 2021, from https://www.turbosquid.com/3d-models/3d-model-main-hall-carnegie-1632884.

[3] Amazon. (n.d.). Amazon S3. Cloud Object Storage | Store & Retrieve Data Anywhere | Amazon Simple Storage Service (S3). Retrieved October 10, 2021, from https://aws.amazon.com/s3/.

[4] Unity Technologies. (n.d.). Unity. Unity Real-Time Development Platform | 3D, 2D VR & AR Engine. Retrieved October 10, 2021, from https://unity.com/.

[5] WebGL. The Khronos Group. (2011, July 19). Retrieved October 10, 2021, from https://www.khronos.org/webgl/.

[6] Unity Learn. How to publish for WebGL. (n.d.). Retrieved October 10, 2021, from https://learn.unity.com/tutorial/how-to-publish-for-webgl.

[7] Amazon. (n.d.). Amazon Relational Database Service (RDS). Amazon RDS | Cloud Relational Database | Amazon Web Services. Retrieved October 10, 2021, from https://aws.amazon.com/rds/.

[8] Amazon. (n.d.). AWS Lambda. AWS Lambda – Serverless Compute - Amazon Web Services. Retrieved October 10, 2021, from https://aws.amazon.com/lambda/.

[9] Django. (n.d.). Retrieved October 10, 2021, from http://www.djangoproject.com/.

[10] React – a JavaScript library for building user interfaces. – A JavaScript library for building user interfaces. (n.d.). Retrieved October 10, 2021, from https://reactjs.org/.

[11] GitHub. (n.d.). Retrieved October 10, 2021, from https://github.com/.

[12] Git. (n.d.). Retrieved October 10, 2021, from https://git-scm.com/.

[13] Unity Asset Store. 3D Animations & Models. (n.d.). Retrieved October 10, 2021, from https://assetstore.unity.com/3d.

[14] 3D Warehouse. (n.d.). Retrieved October 10, 2021, from https://3dwarehouse.sketchup.com/?hl=en.

[15] Blender.org, Blender – Free and Open 3D Creation Software. (n.d.). Retrieved October 10, 2021, from https://www.blender.org/.

[16] AWS Free Tier. Free Cloud Computing Services - AWS Free Tier. (n.d.). Retrieved October 10, 2021, from https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free+Tier+Types=%2Aall&awsf.Free+Tier+Categories=%2Aall&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all.

[17] Docker. Empowering app development for developers. (n.d.). Retrieved October 10, 2021, from https://www.docker.com/.

[18] AWS Lambda Features. AWS Lambda – Product Features . (n.d.). Retrieved October 10, 2021, from https://aws.amazon.com/lambda/features/.

[19] All3DP. OBJ file format – simply explained. (2021, September 21). Retrieved October 10, 2021, from https://all3dp.com/1/obj-file-format-3d-printing-cad/.

[20] Sonia Schechter. (2020, March 19). Everything you need to know about using FBX files. Marxent. Retrieved October 10, 2021, from https://www.marxentlabs.com/fbx-files/.

[21] The Khronos Group. GlTF - runtime 3D Asset delivery. (2020, December 3). Retrieved October 10, 2021, from https://www.khronos.org/gltf/.