



# TED UNIVERSITY

Faculty of Engineering

Departments of Computer & Software Engineering

Project Name: **DiscoVeR**

## **Analysis Report**

### **Team Members:**

CMPE - Levent Bektemur 13759162240

CMPE - Dilge Nisa Üstüntaş 13117245554

SENG - Süzanna Nebioğlu 25123817076

### **Supervisor:**

Ulaş Güleç

### **Jury Members:**

Tansel Dökeroglu

Emin Kuğu

### **Link to the web page of the project:**

<https://humanfromnowhere.github.io/discover.github.io/>

# **Table of Contents**

<b>1. Introduction</b>	<b>3</b>
<b>2. Proposed System</b>	<b>3</b>
2.1 Overview	4
2.2 Functional Requirements	4
2.2.1 User Account and Authentication	4
2.2.2 3D Model Upload and Management	4
2.2.3 VR Scene Conversion and Generation	4
2.2.4 VR Scene Interaction	4
2.3 Non-Functional Requirements	4
2.3.1 Usability	4
2.3.2 Performance	4
2.3.3 Security	4
2.3.4 Reliability and Scalability	5
2.4 Pseudo Requirements	5
2.5 System Models	5
2.5.1 Scenarios	5
2.5.2 Use Case Model	5
2.5.3 Object and Class Model	6
2.5.4 Dynamic Models	6
2.5.5 User Interface - Navigational Paths	7
<b>3. Glossary</b>	<b>8</b>
<b>4. References</b>	<b>11</b>

## **1. Introduction**

Industries like tourism and hospitality etc. heavily depend on visual presentation to attract customers. However, current methods offered often lack interactivity or require users to download specific applications to view 3D environments. "DiscoVeR" aims to bridge this gap by creating a web-based Virtual Reality (VR) touring tool.

Our project aims to provide a web based platform where users (such as hotel owners or real estate agents) can upload their own 3D models (in .fbx or .obj formats). The system processes these models and converts them into interactive VR scenes. These scenes are accessible via a standard web browser using WebXR technology, which allows end-users to explore environments freely using a VR headset without the need to install any external software. Our solution allows access using simple URLs, making the production and consumption of VR tours more democratic.

## **2. Proposed System**

### **2.1 Overview**

The two major user roles for the proposed web-based platform are the Content Creator (Uploader) and the Viewer (End-User). The system takes in raw 3D model files, verifies them, and renders a scene compatible with the WebXR Device API through server-side processing. The frontend will be

lightweight and responsive, ensuring that this is a seamless (high frame rate) and accessible VR experience.

## 2.2 Functional Requirements

### 2.2.1 User Account and Authentication

**Registration:** Users must register with an email and secure password.

**Authentication:** Secure login mechanism to access the dashboard.

**Password Recovery:** Mechanism to reset lost credentials.

### 2.2.2 3D Model Upload and Management

**File Support:** Logged in users can upload 3D models in .fbx and .obj formats.

**Validation:** Once the upload is done the system validates file integrity and security.

**Management:** Users are allowed to view a list of uploaded models and delete them.

**Feedback:** The system provides real-time status updates (success/failure) during upload.

### 2.2.3 VR Scene Conversion and Generation

**Automated Conversion:** Models which are successfully uploaded are processed into a web-viewable VR scene automatically.

**URL Generation:** The system generates a unique, shareable URL for every converted scene.

**Error Handling:** Informative error messages are displayed if conversion fails.

### 2.2.4 VR Scene Interaction

**WebXR Support:** VR Scenes are accessible via browsers supporting WebXR.

**Free-Roam:** Users have the ability to navigate the 3D environment using first-person controls and VR controllers.

**Interface:** A basic user-friendly in-VR menu allows users to re-center the view or exit.

## 2.3 Non-Functional Requirements

### 2.3.1 Usability

The upload interface must be intuitive, minimizing the steps required to generate a VR link.

No external plugins or app installations should be required for the Viewer.

### 2.3.2 Performance

**Frame Rate:** The VR experience must target at least 60 FPS to prevent motion sickness.

**Loading Time:** Generated scenes should achieve initial rendering in under 5 seconds.

### **2.3.3 Security**

**Data Protection:** User passwords must be hashed for security.

**Privacy:** Models which are uploaded must be private unless shared via the generated link.

**Sanitization:** Inputs must be sanitized to prevent XSS and malicious file uploads.

### **2.3.4 Reliability and Scalability**

**Uptime:** Target annual uptime of 99.9%.

**Scalability:** Architecture must support concurrent users and large file processing without the degradation of the server's response time and quality.

## **2.4 Pseudo Requirements**

**Development Methodology:** Agile Software Development with iterative sprints.

**Platform:** Web-based (Desktop for upload, VR/Web-capable device for viewing).

**Libraries:** Three.js for 3D rendering, WebXR API for VR integration.

**Version Control:** Git and GitHub.

**Hardware Constraints:** Server-side processing requires efficient resource usage; End-users are required to have VR headsets for the full experience.

## **2.5 System Models**

### **2.5.1 Scenarios**

#### **Scenario 1: Uploading a Hotel Model**

**Actor:** Hotel Owner (User)

**Flow:** The user logs in and navigates to the "Upload" page. They select a .obj file of their hotel lobby. The system validates the file format and size. Upon success, the system processes the file. The user receives a notification that the "Lobby VR Tour" is ready and is provided a unique link.

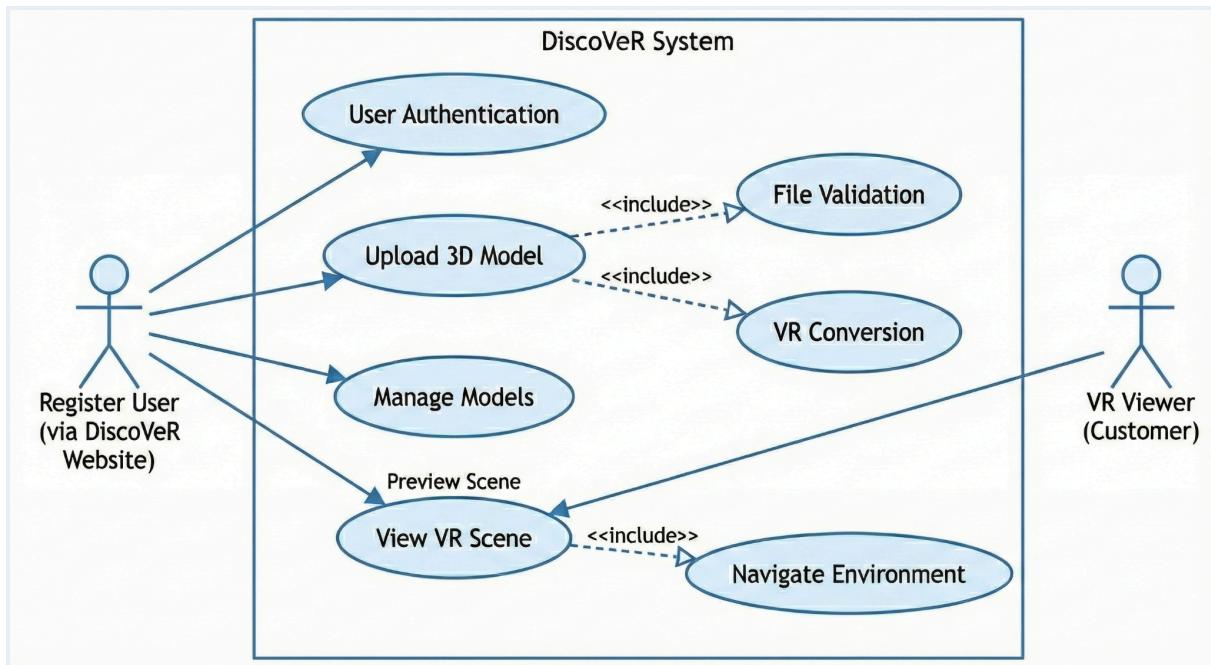
#### **Scenario 2: Viewing the VR Tour**

**Actor:** Customer (Viewer)

**Flow:** The customer follows through the link provided by the hotel owner. The browser opens the 3D scene. The customer puts on their VR headset and clicks "Enter VR". They walk around the virtual lobby using their controllers.

### **2.5.2 Use Case Model**

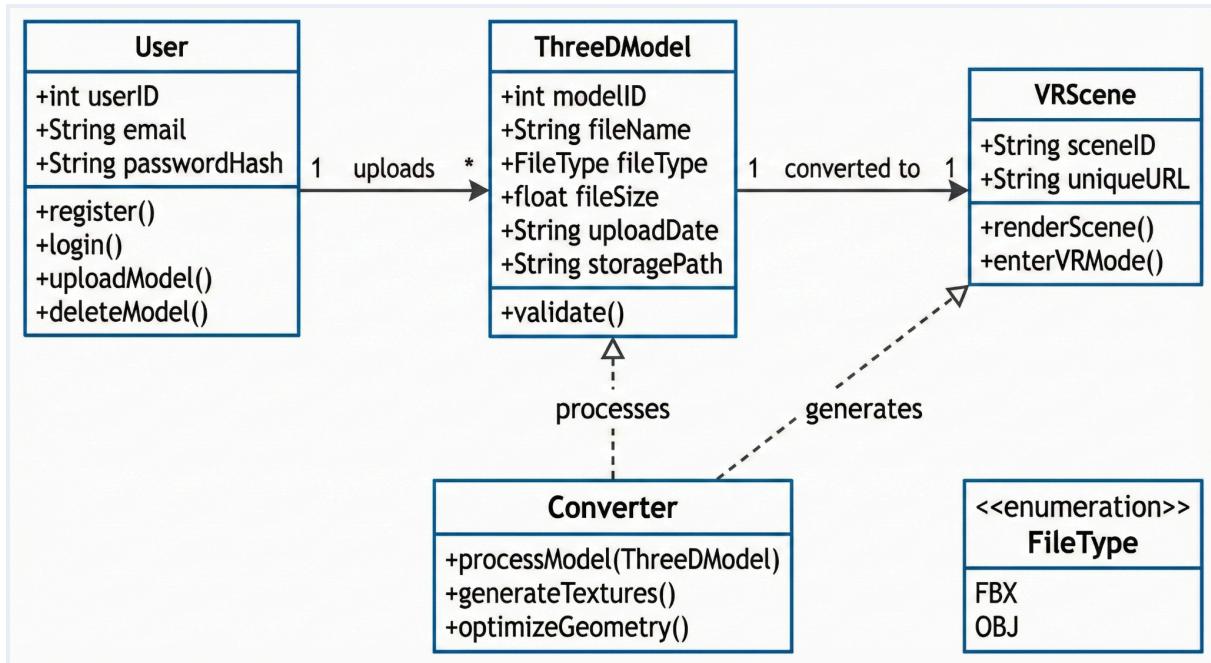
This diagram illustrates the interactions between the User, the Viewer, and the System, covering actions like authentication and model management.



**Figure 1:** Use Case Diagram illustrating the interactions between the User, Viewer, and the DiscoVeR system.

### 2.5.3 Object and Class Model

This model presented below represents the static structure of the system, including the relationships between Users, Models, and the Conversion logic.

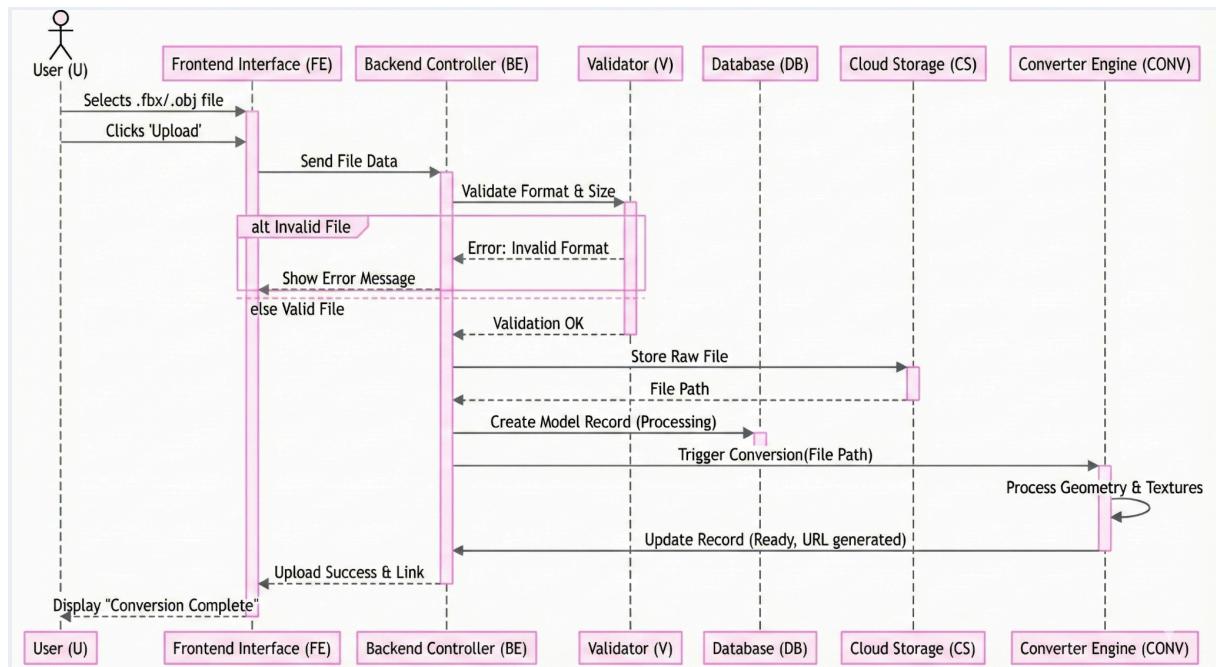


**Figure 2:** Object and Class Diagram defining the system's static structure and relationships between Users, Models, and Scenes

### 2.5.4 Dynamic Models

Sequence Diagram: 3D Model Upload and Conversion

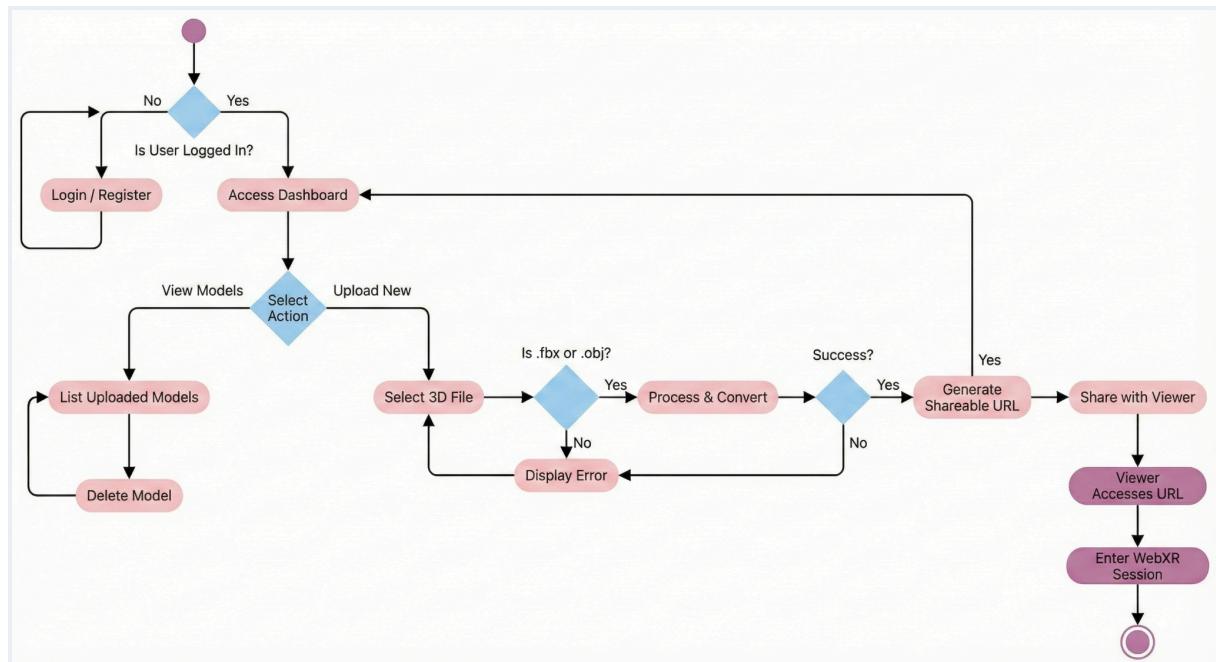
This diagram illustrates the flow of events when a user uploads a new 3D model file.



**Figure 3:** Sequence Diagram detailing the data flow during the 3D model upload and conversion process.

#### Activity Diagram: General User Flow

This diagram represents the overall logic flow from a user perspective.



**Figure 4:** Activity Diagram depicting the general user logic flow from authentication to VR content generation.

#### 2.5.5 User Interface - Navigational Paths

The user interface is designed to be minimalistic and focused on the content (the 3D models).

**Landing/Login Page:** The entry to the system. Offers "Sign In" and "Sign Up" forms.

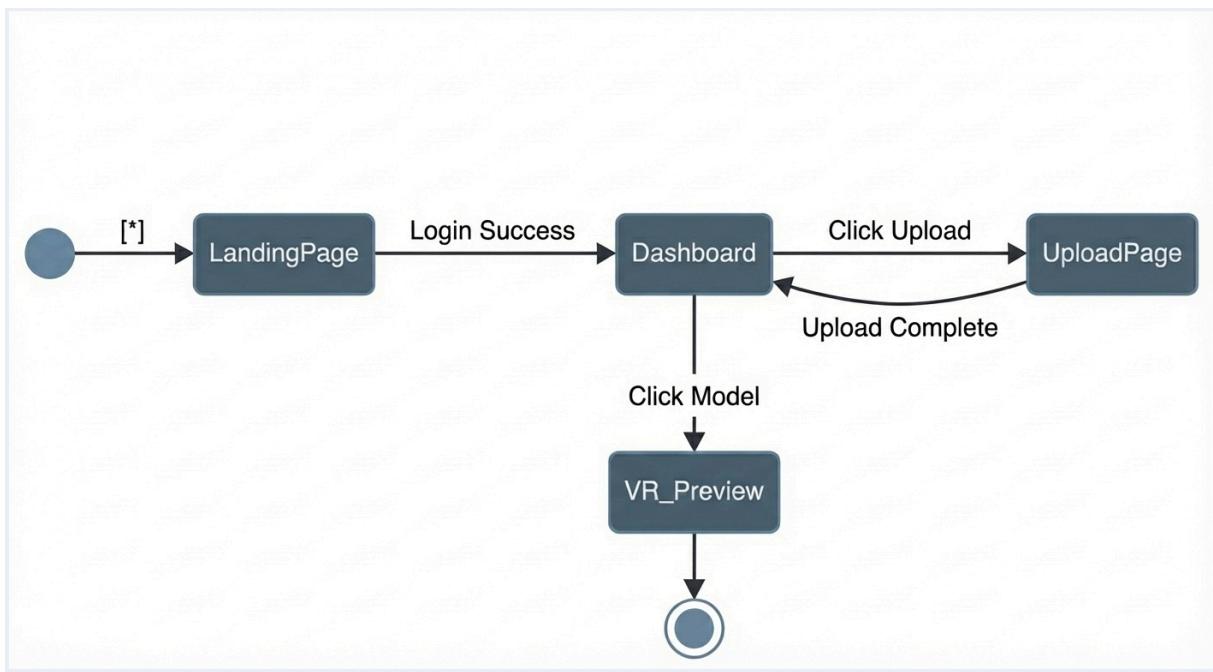
**Dashboard:** The main page. Displays a grid or list of previously uploaded models with their status and links. Contains a prominent "Upload New" button.

**Upload Modal/Page:** A drag-and-drop area for files. Displays progress bars during upload and processing.

**VR View Page:** The public-facing page generated by the system. It contains the WebGL canvas and a button to "Enter VR".

#### Navigational State Diagram:

This diagram depicts the navigational paths and transitions between different user interface screens, such as moving from the Dashboard to the Upload Page



**Figure 5:** Navigational State Diagram showing the transitions between different user interface screens.

### 3. Glossary

**WebXR:** An API for web browsers that allows access to virtual reality and augmented reality devices.

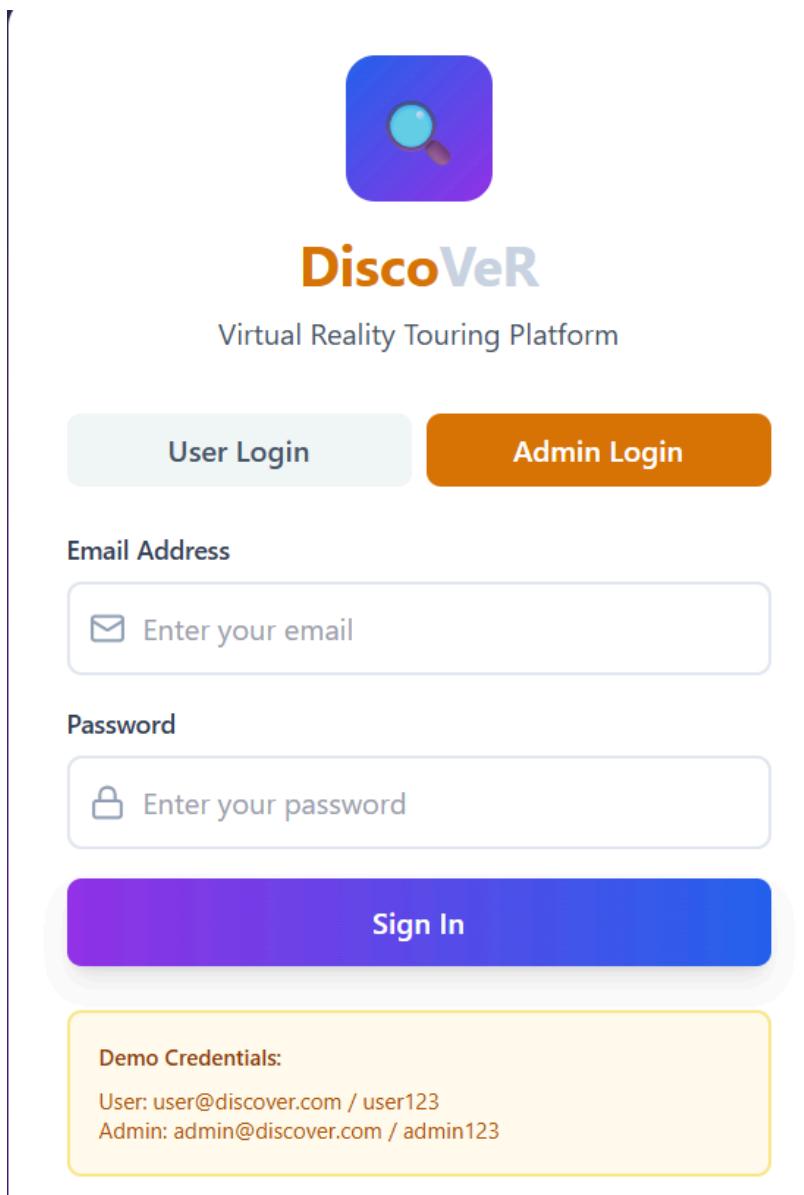
**Three.js:** A cross-browser JavaScript library and application programming interface used to create and display animated 3D computer graphics in a web browser.

**FBX (Filmbox):** An exclusive file format (.fbx) used to provide interoperability between digital content creation applications.

**OBJ:** A geometry definition file format first developed by Wavefront Technologies.

**FPS (Frames Per Second):** Frequency at which consecutive images appear on a display.

**Login Page:** This interface serves as the entry point for the platform, allowing users and admins to securely sign in.



**Figure 6:** Login interface allowing secure authentication for Users and Administrators.

**User and Admin Dashboard:** This screen acts as the central hub where users can view summary statistics about their models and access recent activity.

The image shows two screenshots of the DiscoVeR User Dashboard. Both screenshots feature a dark header bar with the DiscoVeR logo, 'User Dashboard', the user's email 'user@discover.com' (marked as 'User'), and a 'Logout' button.

The top screenshot displays a 'Welcome back!' message and three summary cards:

- Total Models**: 1 (purple card)
- Ready Scenes**: 1 (blue card)
- Processing**: 0 (orange card)

The bottom screenshot also displays a 'Welcome back!' message and three summary cards, identical to the first:

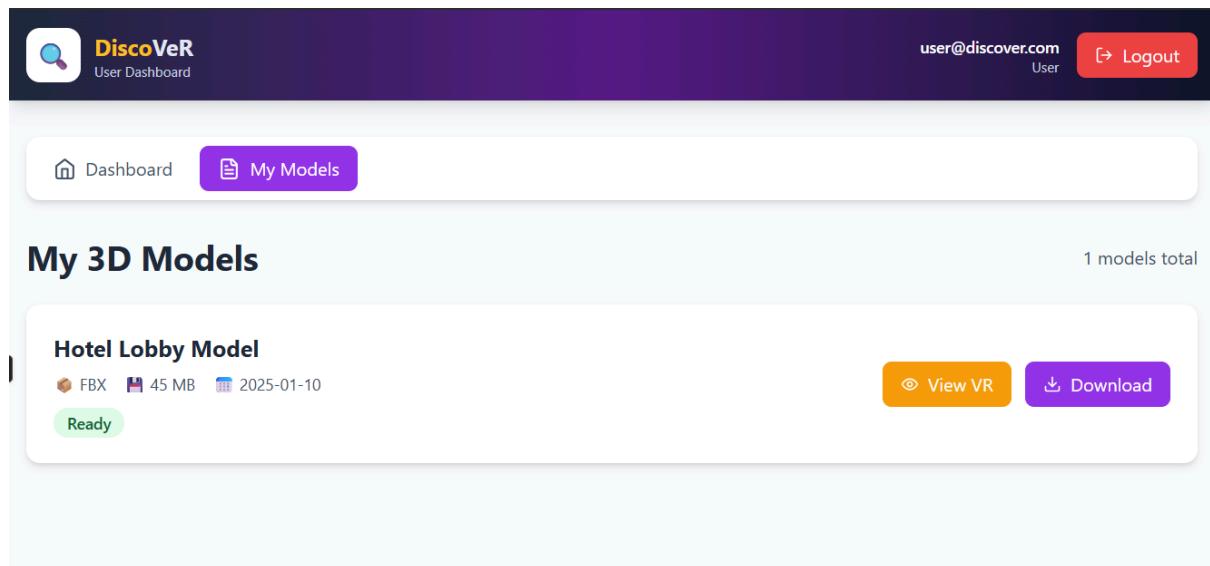
- Total Models**: 1 (purple card)
- Ready Scenes**: 1 (blue card)
- Processing**: 0 (orange card)

Below these cards is a section titled 'Recent Activity' containing a single item:

**Hotel Lobby Model**  
Uploaded on 2025-01-10 Ready

**Figure 7:** User Dashboard displaying model statistics, recent activity, and the upload option.

**My 3D Models:** This page lists the user's uploaded content, displaying file details and status while providing options to view the VR scene or download the file.



**Figure 8:** "My Models" page showing the list of uploaded files, conversion status, and VR viewing options.

## 4. References

**W3C.** (2024). *WebXR Device API Specification*. Retrieved from <https://www.w3.org/TR/webxr/>

**Khronos Group.** (2021). *gltf 2.0 Specification*. Retrieved from <https://www.khronos.org/gltf/>

**Autodesk, Inc.** *Autodesk FBX SDK*.

**IEEE.** (2020). *IEEE Code of Ethics*.