



Friday, April 26, 2024



Applied Software Engineering Capstone

# Grace

An open-source, cross-platform collection management platform



**Presented By:**  
Michael Muzzarelli

**Presented To:**  
Dr. Samuel Cho



# Introduction

Grace as a capstone project is split into three primary components:

## Client

# 1

Grace is cross-platform client application, built in Dart with Flutter

## Server

# 2

Grace Authentication API is a Node-based REST API serving authentication requests

## Server

# 3

Grace Material API is a Go-based REST API serving material requests



# Grace

Grace is available on iOS and web, supporting the following features:

- sign up and sign in with an email address or with Google
- browse books, games, and movies and add them to your user library
- manage and update your user profile

Additionally, as a Grace administrator:

- manage registered users and their role and status via administrative dashboard



# Grace Authentication API

Authentication API is accessible with REST, supporting the following features:

- account sign-up with email and password or Google.
- account sign-in with email and password or Google.

Changes are automatically made available with GitHub Action CI/CI:

- Linting and test suites are executed and semantic version is created
- Build is generated and redeployed with Heroku



# Grace Material API

Material API is accessible with REST, supporting the following features:

- fetch, store, and update books, games, and movies based on:
  - title, franchise, author, etc.
  - generated or supplied identifier (e.g., ISBN)

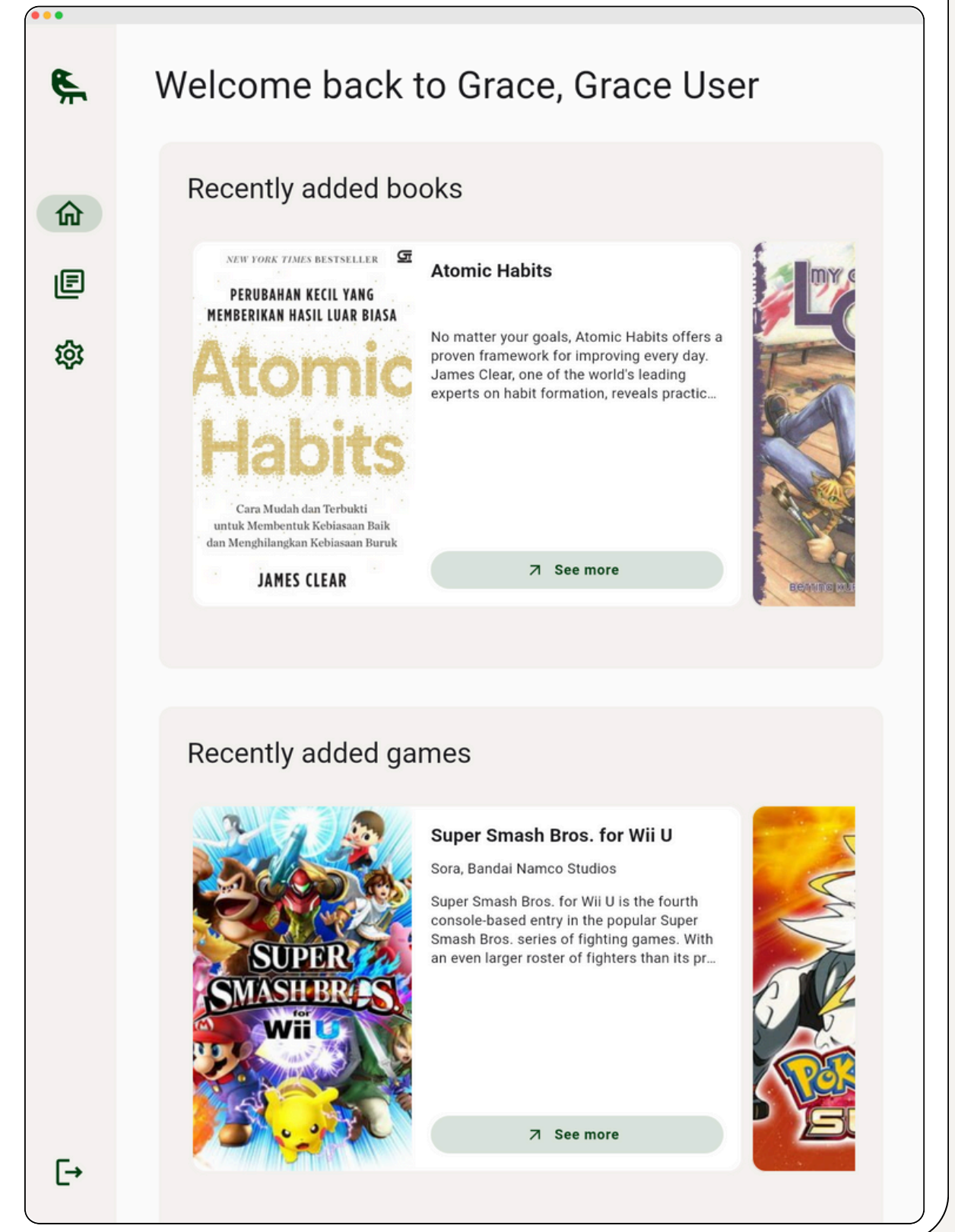
Changes are automatically made available with GitHub Action CI/CI:

- Linting and test suites are executed and semantic version is created
- Build is generated and redeployed with Heroku

# Browse

Grace enables users to browse books, games, and movies and add them to their personal library.

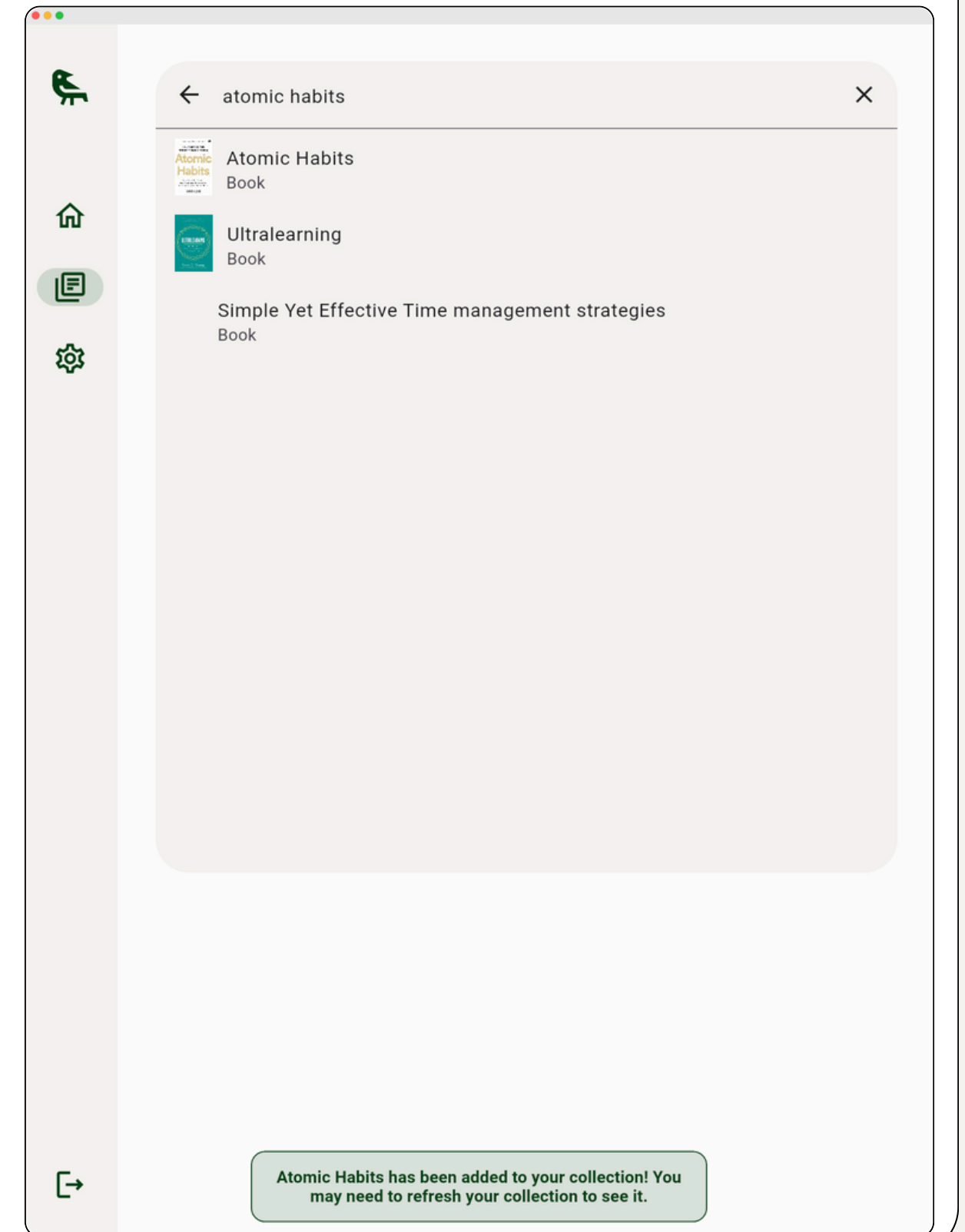
On the home screen, materials recently added to Grace are displayed to encourage this browsing, and each material in this view can be added to a personal library.



# Search

Grace also enables material search on the library screen, searchable by title, franchise, author, generated identifier (e.g., ISBN), and many more.

Each search result can be selected to add to the Grace platform and add to the user personal library.



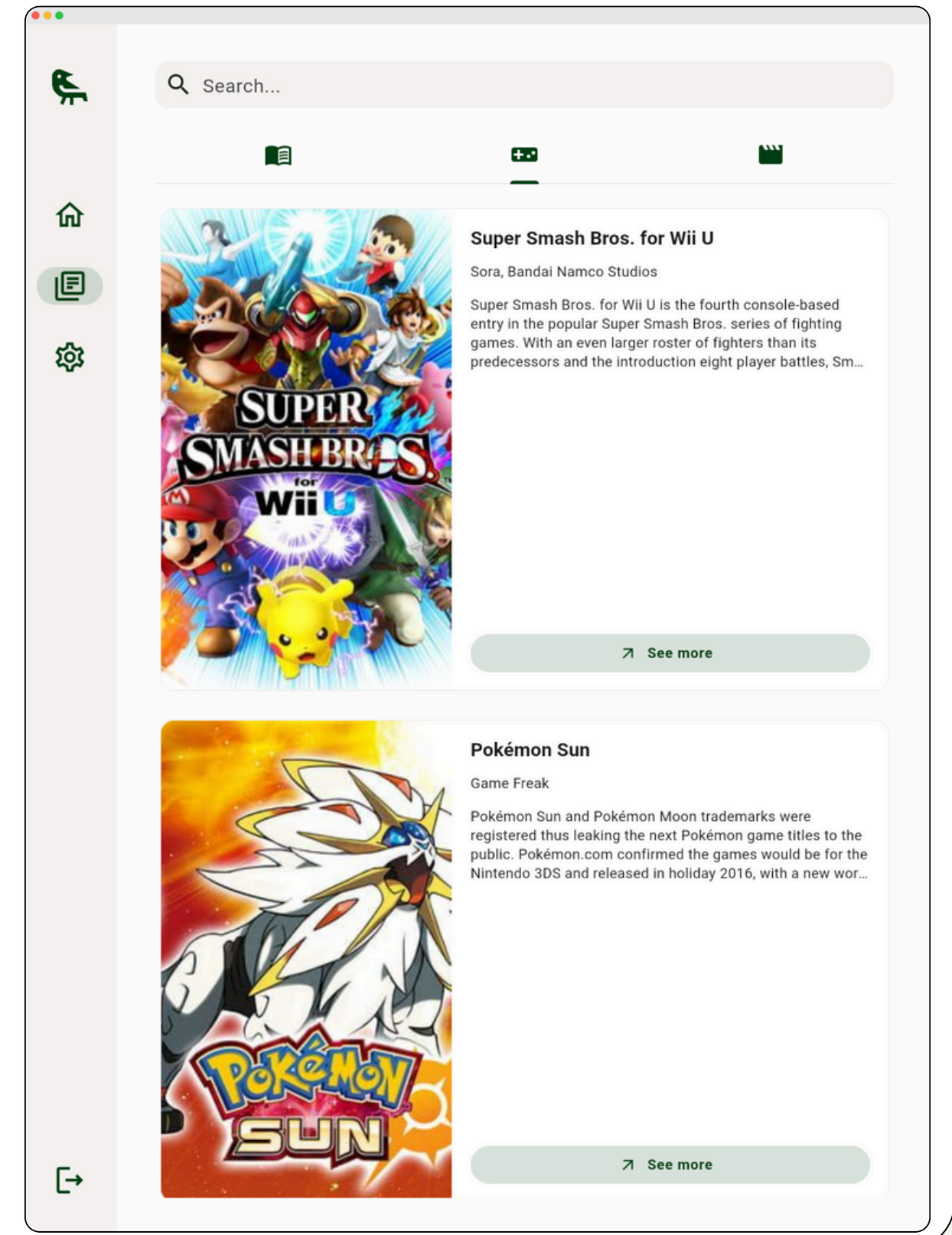




# Collect

Once a user adds a material to their personal library, it will show up in the respective tab on the library screen; books, games, or movies.

This provides a condensed, easily navigable view of all materials a user has added to their personal library.

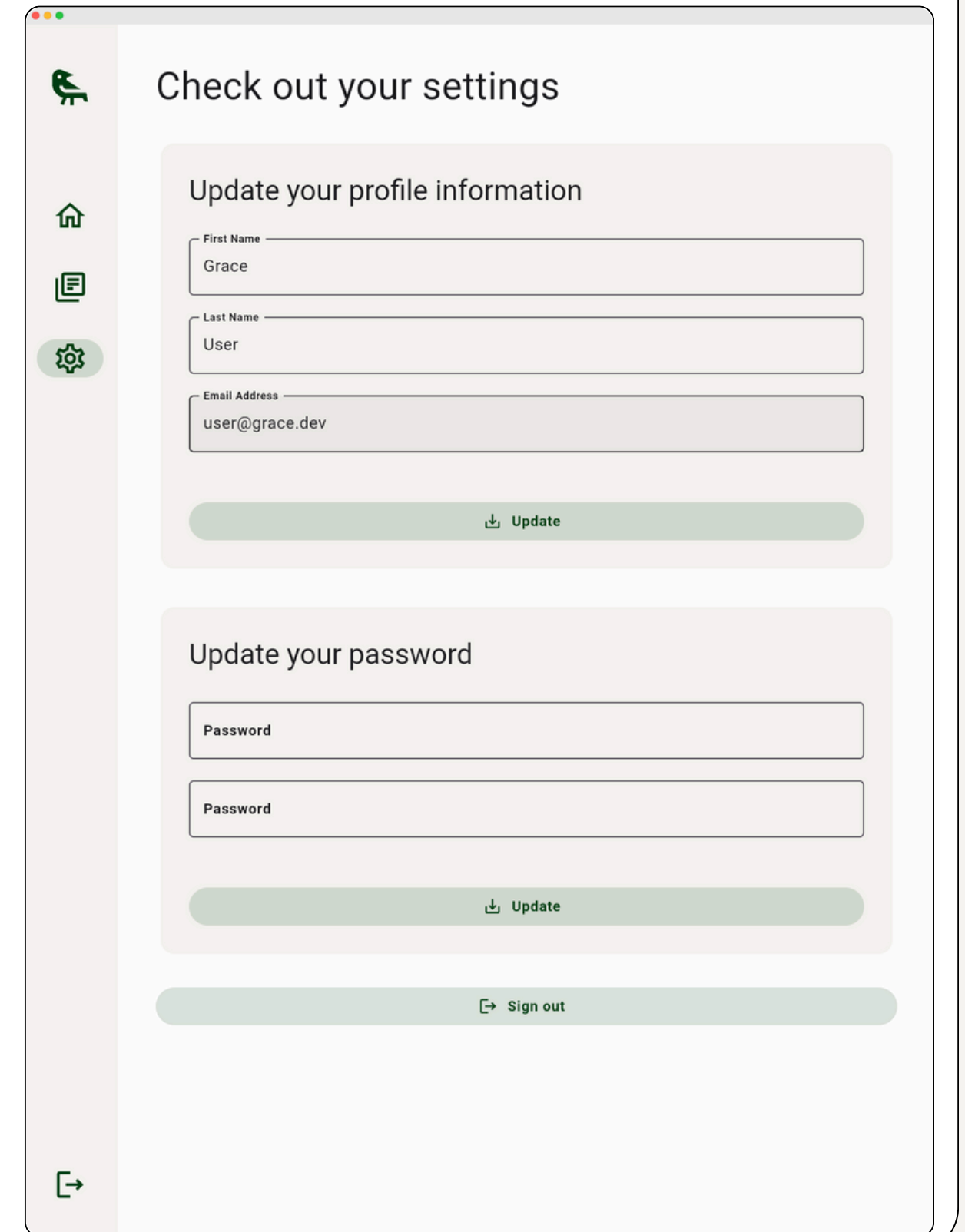




# Manage

Users can manage their profile and sign-in information from the user profile dashboard.

This enables users to update an email address, name, or password, and many more properties as we incorporate them in the platform.



A mockup of a user profile dashboard. It features a sidebar with icons for home, list, and settings (the settings icon is highlighted). The main content area is titled 'Check out your settings' and contains two sections: 'Update your profile information' and 'Update your password'. The first section has input fields for First Name (Grace), Last Name (User), and Email Address (user@grace.dev), followed by an 'Update' button. The second section has two 'Password' input fields, followed by an 'Update' button. At the bottom of the main area is a 'Sign out' button. The entire interface is enclosed in a window frame with three dots in the top-left and a close button in the top-right.

Check out your settings

Update your profile information

First Name  
Grace

Last Name  
User

Email Address  
user@grace.dev

Update

Update your password

Password

Password

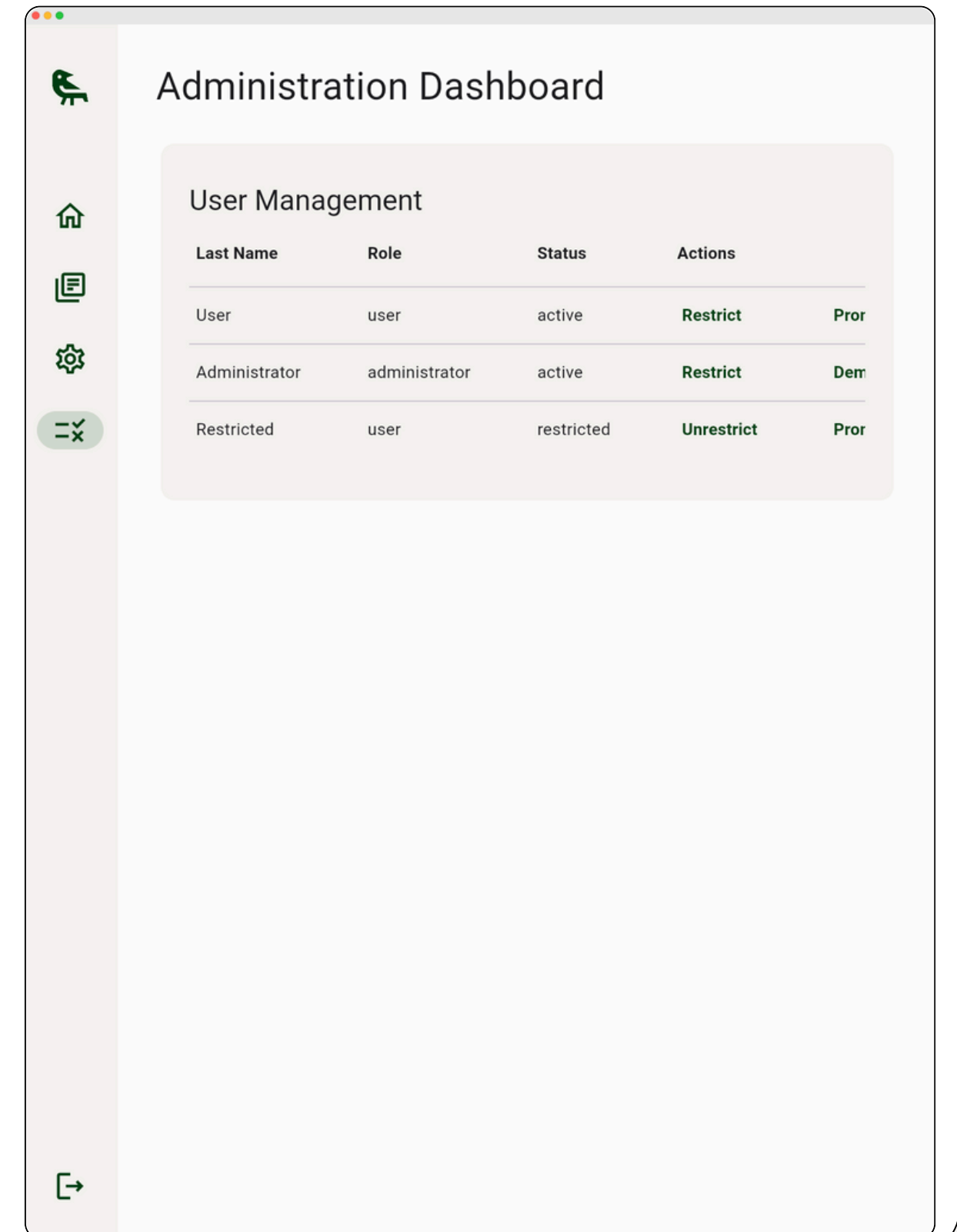
Update

Sign out

# Moderate

Platform administrators can moderate users on the administrative dashboard.

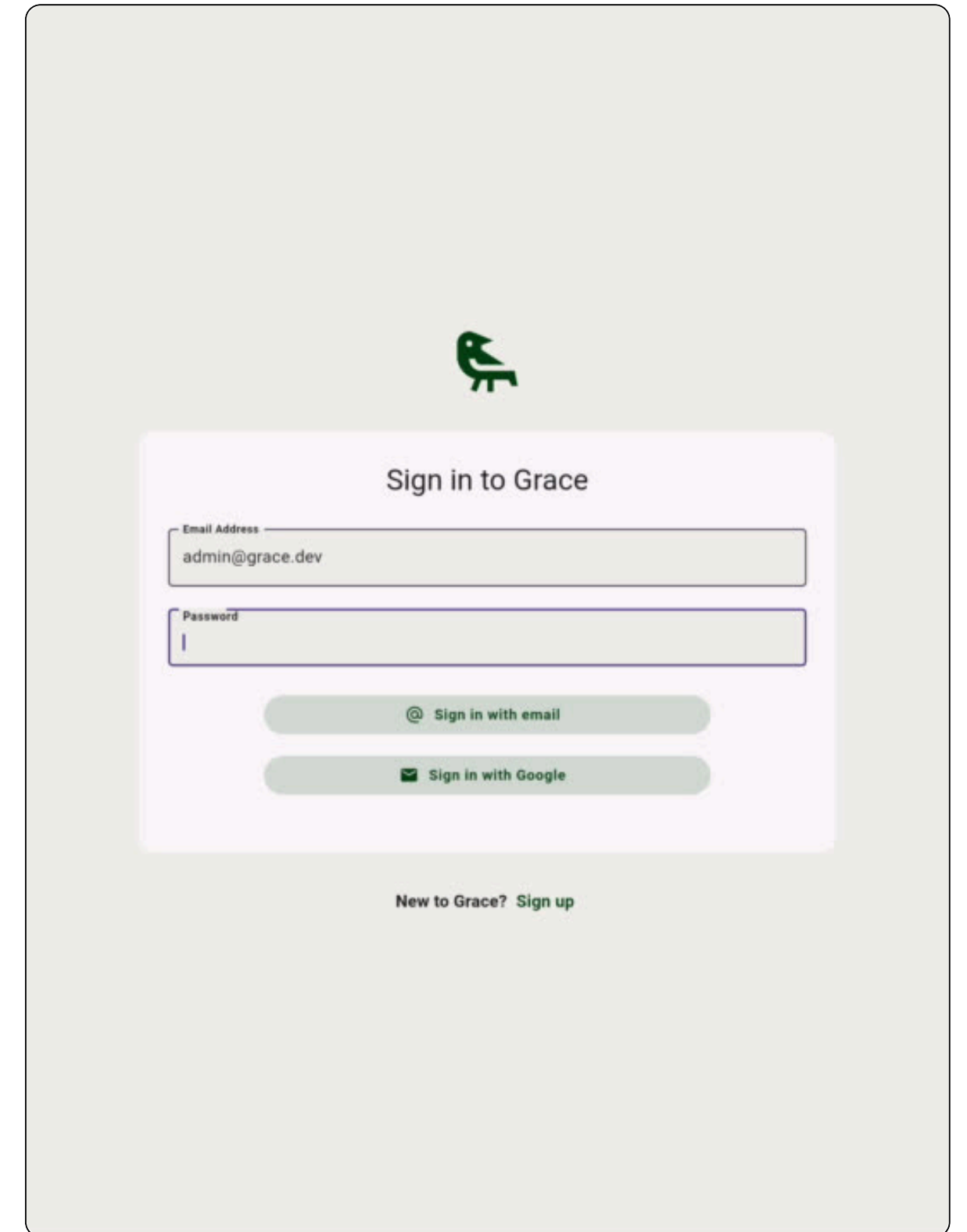
This enables administrators to view, restrict or unrestrict, and promote or demote registered users.



# Demonstration

The recording on the right demonstrates a common experience within Grace.

In this experience, an administrator signs in, searches a material and adds to their library, manages their profile, and views the administrative dashboard.





# Artifacts

Final iteration includes the following artifacts:

**Grace**

**# 1**

Available [here](#) - interfaces with Authentication API and Material API via in-platform services.

**Auth API**

**# 2**

Available [here](#) - interfaced with Grace, versioned and deployed to Heroku with GitHub Action CI/CD.

**Material API**

**# 3**

Available [here](#) - interfaced with Grace, versioned and deployed to Heroku with GitHub Action CI/CD.



# Closing thoughts

I cannot express how much I appreciate all I learned throughout this process:

- CI/CD with GitHub Actions
- deployment on Heroku with custom domain, SSL, etc.
- appropriate state management and the significant impact without it

Most important, what I would do different knowing what I do now:

- dependency injection is important to work into design as early as possible.
- buy, don't build, is the **most** important rule.