

print(d)

3D Printing made easy

CHANGE HISTORY

| VERSION | SUMMARY | AUTHOR | DATE |
|---------|--|--------|------------|
| 0.1 | Initial Draft | All | 02/05/2017 |
| 0.2 | Added project description and requirements | All | 02/09/2017 |
| 0.3 | Diagrams | All | 02/23/2017 |
| 0.4 | Activity Diagrams | All | 02/28/2017 |
| 1.0 | First Official version | All | 03/02/2017 |
| 1.1 | Final | All | 04/28/2017 |

TABLE OF CONTENTS

| | |
|---|----|
| CHANGE HISTORY | 1 |
| TABLE OF CONTENTS | 2 |
| 1. INTRODUCTION | 4 |
| 1.1 Document Purpose | 4 |
| 1.2 Scope | 4 |
| 1.3 Definitions | 4 |
| 2. PROJECT DESCRIPTION | 6 |
| 2.1 Product Purpose | 6 |
| 2.2 Product Features | 6 |
| 2.2.1 Android Application | 6 |
| 2.2.2 Thingiverse API | 6 |
| 2.2.3 Slicing Server | 6 |
| 2.2.4 Octoprint Server | 6 |
| 2.3 Constraints | 6 |
| 2.4 Assumptions & Dependencies | 6 |
| 3. FUNCTIONAL REQUIREMENTS | 8 |
| 3.1 User Account Management | 8 |
| 3.2 Setting up printer profile within app | 8 |
| 3.3 Pulling collections from Thingiverse | 8 |
| 3.4 Remote server (Heroku) | 8 |
| 3.5 Pushing to Octoprint Server | 9 |
| 3.6 Filament Tracking | 9 |
| 3.7 Alignment / Bed Level | 9 |
| 3.8 Error Handling | 9 |
| 4. NON-FUNCTIONAL REQUIREMENTS | 10 |
| 4.1 Hardware Requirements | 10 |
| 4.2 Location Requirements | 10 |
| 4.3 Service Requirements | 10 |
| 4.4 Usability Requirements | 10 |
| 4.5 API Requirements | 10 |
| 4.6 Backend Server Requirements | 10 |
| 5. UML DIAGRAMS | 11 |
| 5.1 Use Case Diagram | 11 |
| 5.2 Class Diagram | 12 |
| 5.3 Activity Diagrams | 13 |

6. APPLICATION SCREEN MOCKUPS

18

1. INTRODUCTION

1.1 Document Purpose

The purpose of this document is to define the formal requirements of the Print(d) application. It outlines the functional and nonfunctional requirements along with the intended features of the application.

1.2 Scope

This project will consist of an Android application that integrates Thingiverse, Octoprint, and a slicing server with a user-friendly interface. Modules include the slicing server, Thingiverse API, Octoprint API, filament tracking, and printer configurations.

In addition to the application, a backend server will be created to allow users to convert digital models to a file format their printer can understand.

1.3 Definitions

- **3D Printing** - A manufacturing process known as Additive Manufacturing, where parts are fabricated through building layer on layer.
- **3D Printer** - A machine that can extrude material in successive layers to produce a part from a digital model.
- **API (Application Program Interface)** - a publicly available interface that provides developers with programmatic access to a proprietary software application.
- **Bed** - The bottom plane onto which the 3D printer extrudes filament.
- **Extruder** - The “hot end” of the printer. A nozzle that reaches high heat to melt filament and position it to form a model.
- **FDM (Fused Deposition Modeling)** - A type of Additive Manufacturing commonly used for many commercially available 3D printers.
- **Filament** - The material used to print with. Usually a roll of 1.75mm PLA plastic.
- **Gcode** - The file type used by 3D printers.
- **Model** - The part that is to be printed.
- **Octoprint** - An open source, internet connected controller for most models of 3D printers. Typically run on a Raspberry Pi or other hobbyist cheap computer.

- **Raspberry Pi** - A small, cost effective hobbyist computer usually used for standalone projects, such as controlling an Octoprint server.
- **Slicing Software** - A program designed to take a STL file model and convert it into Gcode that a 3D printer can read.
- **Thingiverse** - A website, created by MakerBot Industries, designed to be a place for people to upload and share models designed to be 3D printed.

2. PROJECT DESCRIPTION

2.1 Product Purpose

We've noticed how difficult it can be to get into 3D printing, so we are designing a better, more user-friendly approach for consumers to be able to learn 3D printing.

Print(d) will allow users to configure their printer and print custom models without ever having to interact with gritty printer details or modeling and slicing software. The software interface will help walk users through previously tedious tasks such as bed calibration and model slicing, all within one centralized space.

2.2 Product Features

2.2.1 Android Application

We will develop a mobile application that connects all of the core functionality. The application will be connected to the user's Print(d) and Thingiverse accounts. Once logged in, the user will be able to select 3D models from Thingiverse and send it a local Octoprint server for 3D printing.

2.2.2 Thingiverse API

Our mobile application will connect the user's Thingiverse account with their Print(d) account. The user will then be able to pull Thingiverse collections to select for printing.

2.2.3 Slicing Server

When the model is ready to be sliced, the user will be able to send that model to our slicing server. We will implement the model slicing functionality using a remote Heroku server.

2.2.4 Octoprint Server

After the model has been sliced, the user will use Octoprint to interface with the desired 3D printer. We will provide basic instructions for the user to set up an Octoprint server (if it is not already set up) on a Raspberry Pi that is connected to the printer.

2.3 Constraints

- Must be on same wifi network as the Raspberry Pi.
- Constrained to just Thingiverse, can't use similar sites.

2.4 Assumptions & Dependencies

- Thingiverse and Octoprint APIs

- The user's devices are wifi enabled and are connected to the internet
- Heroku online

3. FUNCTIONAL REQUIREMENTS

3.1 User Account Management

| DESCRIPTION | PRIORITY |
|------------------|----------|
| Account Creation | High |
| Login | High |
| Logout | Low |
| Delete Account | Low |

3.2 Setting up printer profile within app

| DESCRIPTION | PRIORITY |
|---------------------|----------|
| Build Volume | High |
| Make/Model | High |
| Heated Bed checkbox | High |

3.3 Pulling collections from Thingiverse

| DESCRIPTION | PRIORITY |
|--|----------|
| Thingiverse User Authentication | High |
| Collection Tracking | High |
| Ability to retrieve a model (.stl) file from a user's collection | High |

3.4 Remote server (Heroku)

| DESCRIPTION | PRIORITY |
|--|----------|
| Heroku backend server will be able to receive a .stl file and utilize the Cura Engine to slice the file into a .gcode file | Medium |
| Server will be able to authenticate users and manage account creation | Medium |

3.5 Pushing to Octoprint Server

| DESCRIPTION | PRIORITY |
|---------------------------------------|----------|
| Connect mobile app to Octoprint API | High |
| Send sliced model to Octoprint server | High |

3.6 Filament Tracking

| DESCRIPTION | PRIORITY |
|----------------------------------|----------|
| Total filament input and storage | Low |
| Track filament per job | Low |

3.7 Alignment / Bed Level

| DESCRIPTION | PRIORITY |
|--|----------|
| Direct control of 3D Printer over Octoprint to move extruder head to necessary positions | Medium |
| Companion interface to walk users through the process | Medium |

3.8 Error Handling

| DESCRIPTION | PRIORITY |
|---------------------------------------|----------|
| Separate error messages for each step | Low |

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Hardware Requirements

- The user must have a 3D printer
- The user must have an Octoprint server connected to their 3D printer
- Octoprint must be connected to the internet
- The user must have an Android device

4.2 Location Requirements

- The user must be connected to the same wifi network as the desired Octoprint server

4.3 Service Requirements

- The user must have a Thingiverse account

4.4 Usability Requirements

- The Print(d) application must be easy to use for users that have little to no experience with 3D printing.

4.5 API Requirements

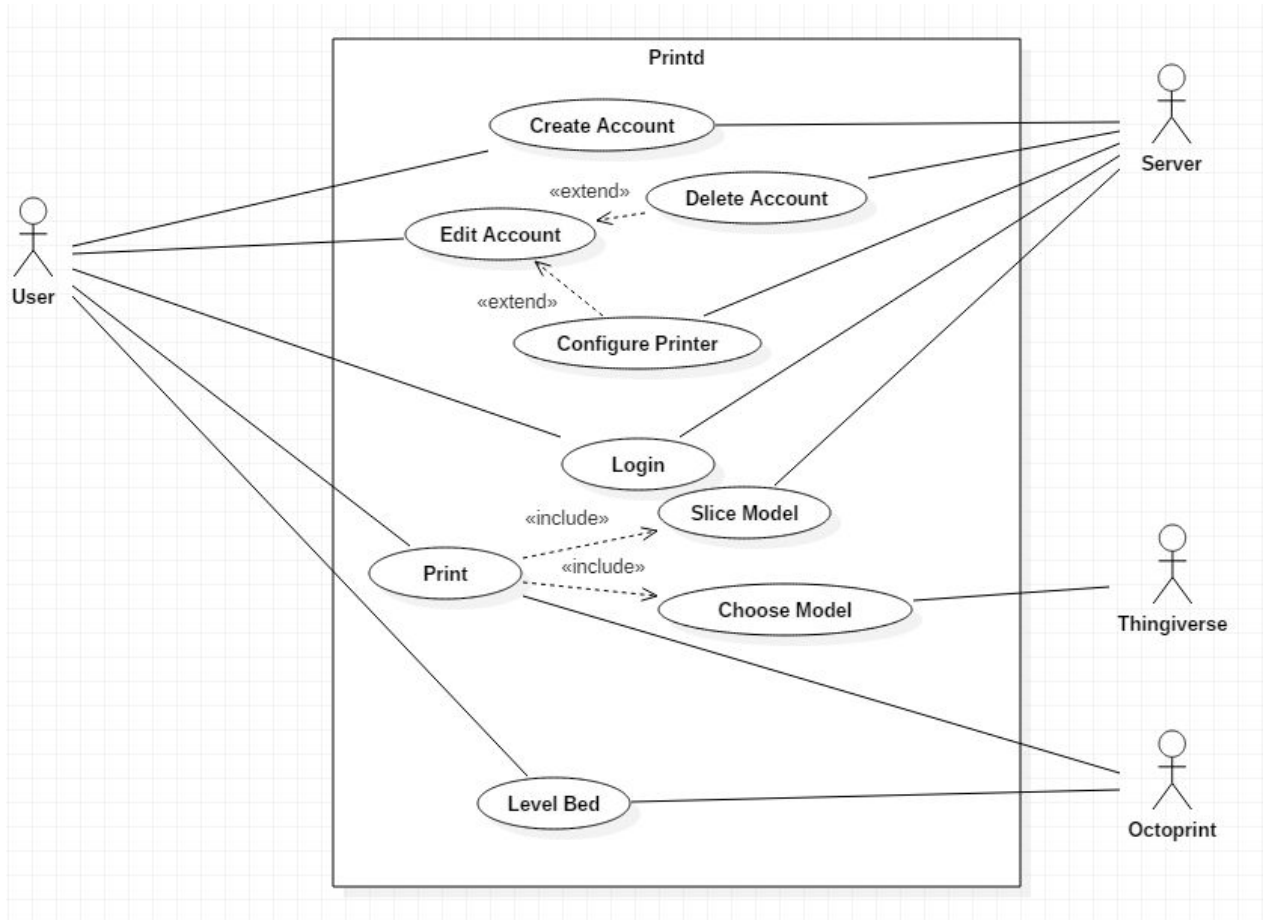
- Thingiverse API open and available
 - Allows users to view their collections
- Octoprint API allows for direct printer control and pushing Gcode files

4.6 Backend Server Requirements

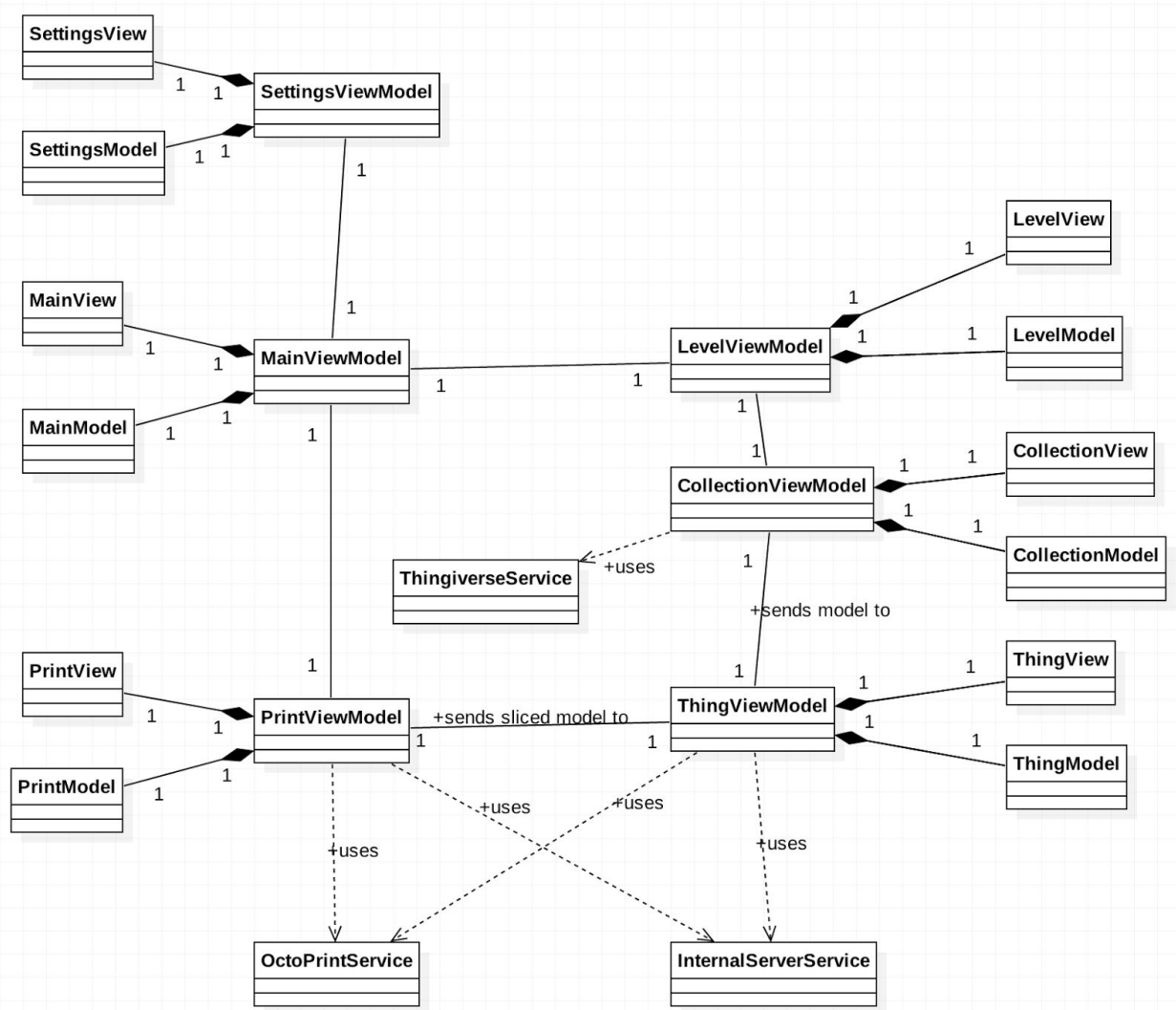
- The server will be able to communicate through an API to the Android application
- The server will be able to store user data
- The server will be integrated with the Cura Engine

5. UML DIAGRAMS

5.1 Use Case Diagram



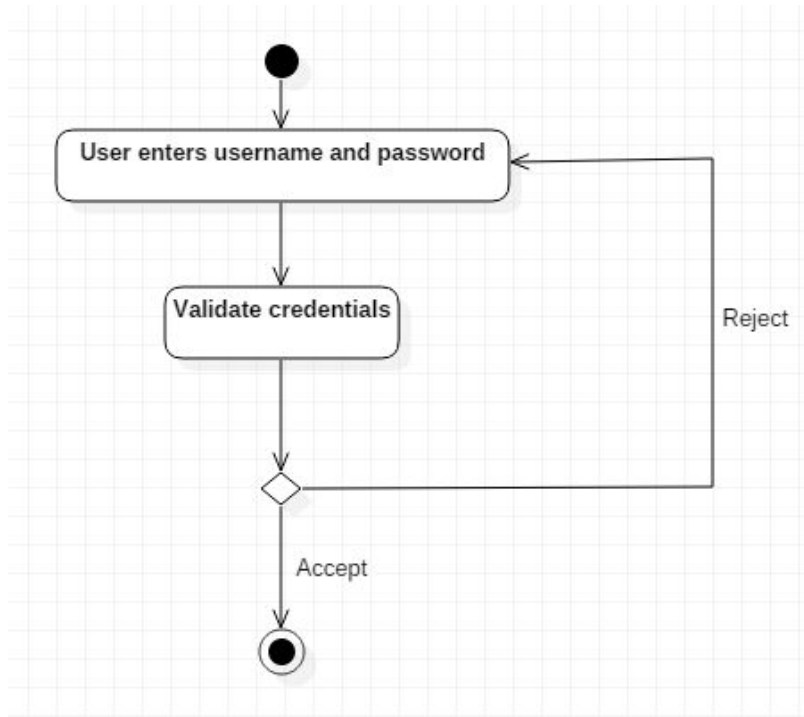
5.2 Class Diagram



There will be three classes per view: the View, Model, and ViewModel. Service classes will connect external services to our Android application.

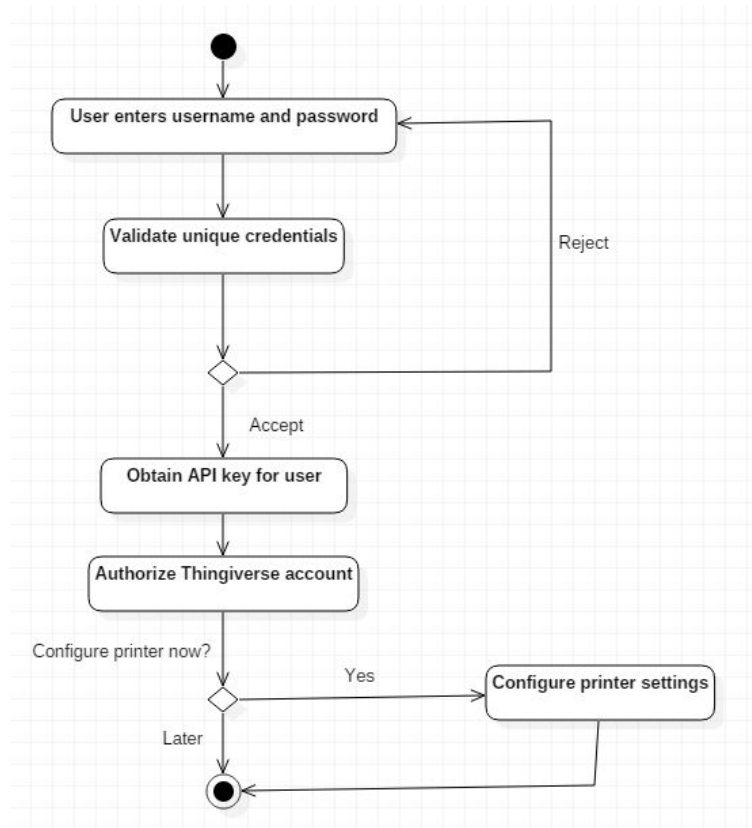
5.3 Activity Diagrams

Login



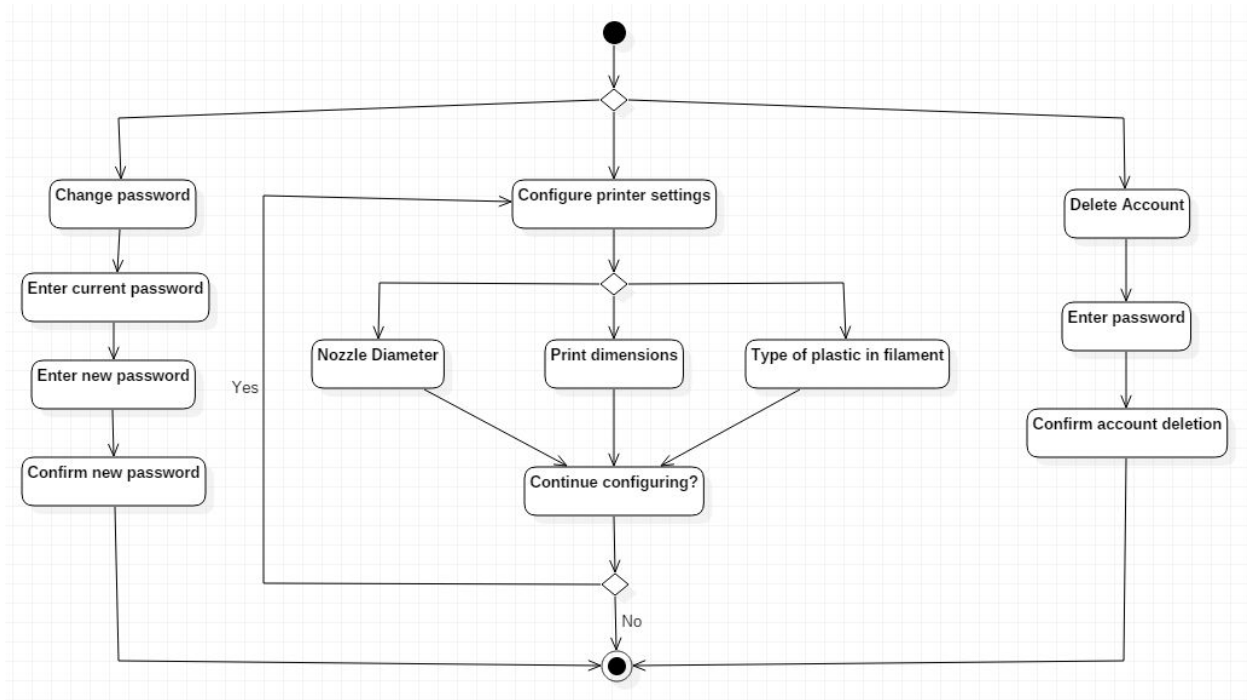
This is a diagram for a generic login activity. The user needs to enter their username and password and hit the login button. We will also have the option to remember their login info and log them in automatically when they launch the app. This could be a security issue so it is only recommended if the user has a passcode on their phone.

Create Account



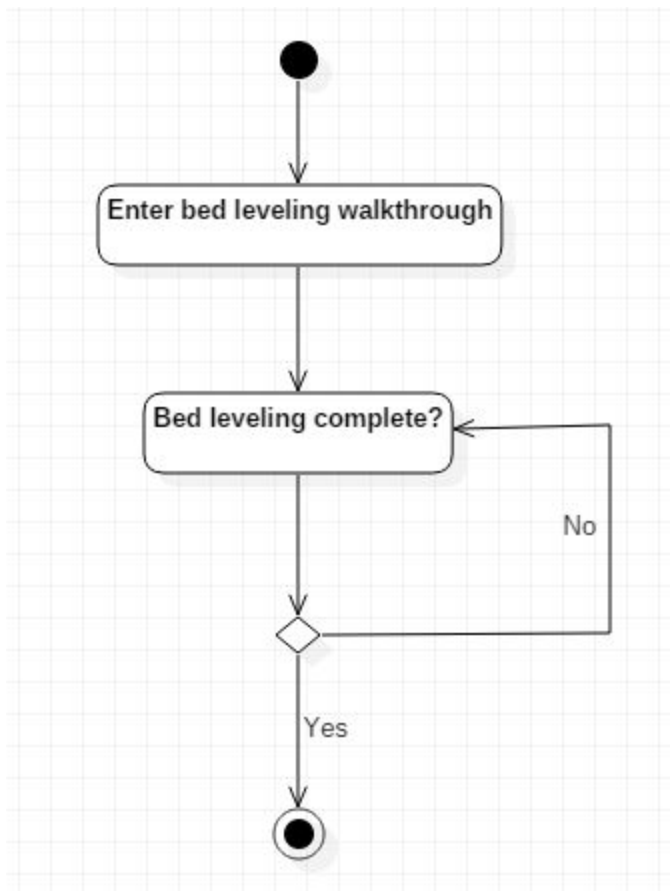
This is the activity diagram for creating an account. After the user creates a unique username and sets a password, they will be prompted to link their Thingiverse account. They are then presented with the option to configure their printer settings now or later. While it isn't necessary to configure the printer settings right away it is recommended.

Edit Account



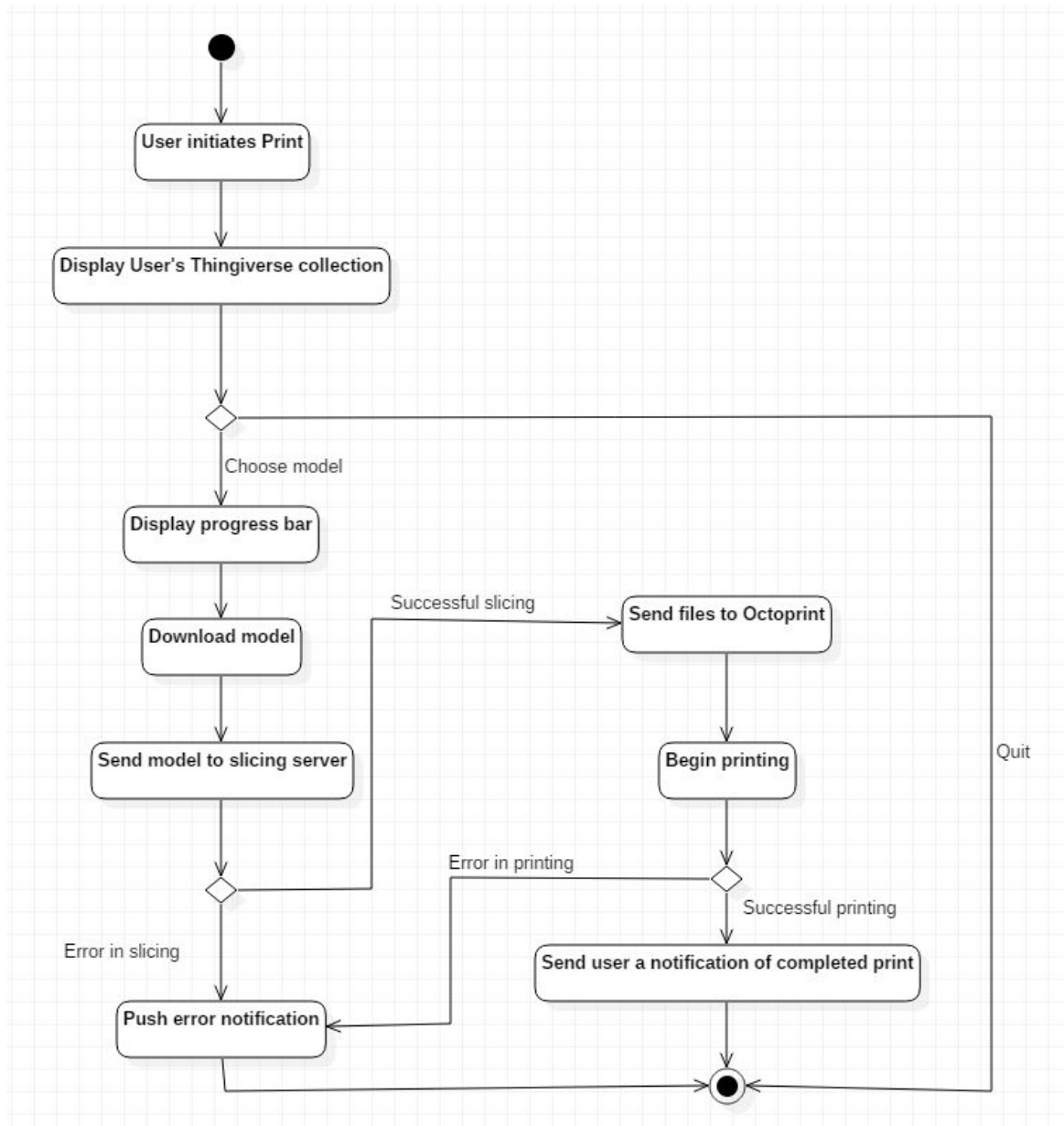
The activity for editing their account is pretty standard. They can do basic account management stuff like change their password or delete their account. The important part of this activity is that they can configure their printer settings here. This lets them put their printer model in, size of the bed, and type of filament. Ideally we will have a database full of printer settings created by users. This way new users can just select which type of printer they have and have all of the settings loaded in.

Bed Levelling



Having app-assisted bed leveling is pretty straightforward on the technical side. It is mostly just an instructional walkthrough on how to level the bed. The process is very tedious and this would help alleviate that.

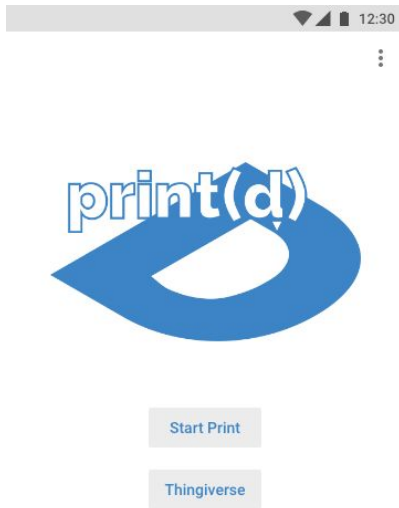
Printing



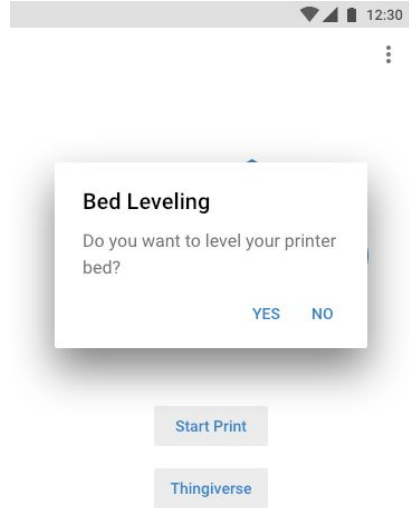
This is the main activity diagram as it covers printing. This highlights how much this app simplifies the process, since the user only has to open the app and choose a model to print. Since there are a lot of steps it is important to have a progress bar. Once the printer starts printing the progress bar can be changed to text saying that the print has begun. Since the printer may be left unattended in another room, the app will send a push notification to the user once printing has completed. Should the print fail at any point in the process, an error will be displayed to the user telling them at what step the app failed. There are many things that can go wrong when printing so it is important to be specific so the user can potentially fix the issue.

6. APPLICATION SCREEN MOCKUPS

Printing Status



Bed Leveling Option



Bed Leveling Instructions

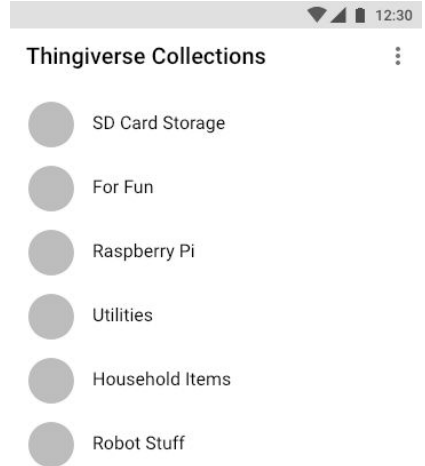


Step 1: Paper Setup

Ethical snackwave bitters hashtag four dollar toast, unicorn food truck venmo fingerstache tousled air plant plaid mustache hammock lomo. +1 vice keffiyeh XOXO venmo. Live-edge cold-pressed ramps, letterpress waistcoat butcher literally pabst lyft.



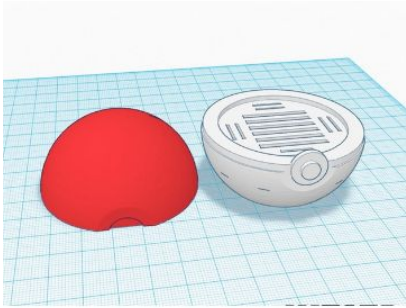
Thingiverse Collection Viewer



Model Viewer

12:30

Model Viewer



Top_part.stl

Bottom_part.stl



Printing Status

12:30

Printing Status



Print is 31% complete

