

Team Name: Pi in the Sky

Team Members:

- David (Andrew) Barlow
- Matt Behrens
- Meghan Donohoe
- Jared Cantilina
- Aparajithan Venkateswaran

Description:

The end goal of this project is to produce a simple *Raspberry Pi* based home server that requires minimal setup. It can be used for a variety of purposes, such as cloud storage, automated backups, web hosting, etc. It will include tools to help the end user set up a home server with minimal technical knowledge on their part.

It will also minimize costs for users by using relatively inexpensive equipment. The total hardware investment will be less than \$70. In the long run, this is much cheaper than other commercial alternatives such as *Amazon Web Services*, *Google Cloud*, etc. It will also be accessible outside the user's home network through a website that will act as an admin dashboard.

Vision Statement: To make server hosting cheaper and accessible.

Motivation:

There is currently a lack of inexpensive home server solutions that don't require extensive IT knowledge to use. We want to create a simple plug and play home server. The idea is that it can be plugged in and not have to deal with setting up a convoluted server, but a simple interface to start using it as soon as possible.

Risks:

- Lack of experience using *Raspberry Pi*
- Initial investment in hardware
- No way to automatically set up port forwarding on end user's router to make server accessible from WWW
- Difficulty setting up networking on university's network

Risk Mitigation Plan:

- Use online resources such as videos, tutorials, etc. to familiarize with *Raspberry Pi*
- Get cheap testing hardware
- Provide the user with a manual that explains how to setup port forwarding in simple language
- SSH into device using outside network

Version Control System: Git and GitHub**Development Method:** Agile**Collaboration Tools:**

1. Slack for communication
2. Trello for task management

Proposed Architecture:

This project's hardware will consist of a *Raspberry Pi*, an external hard drive, and SD card.

The backend software will consist of Apache, MySQL, PHP, OwnDrive, etc. which will be run on Linux. But this is subject to change as we start building the project and realize that some options are better suited for our needs than others.

The front-end software is not completely locked down, and is highly dependent on what we decide to use as the default functionality of this server. The plan right now is to have a website, written in HTML, CSS, and JavaScript, that will act as the admin dashboard. Once again, this is subject to change as we progress towards the end goal.