Contents

1	Analysis			
	1.1	Problem Identification	2	
		1.1.1 Problem Description	2	
		1.1.2 Interview	2	
		1.1.3 Existing similar solutions	2	
		1.1.4 Features to be incorporated into solution	4	
		1.1.5 Feedback from stakeholders	4	
	1.2	Requirements	4	
		1.2.1 Stakeholder requirements	4	
		1.2.2 Software and hardware requirements	4	
		1.2.3 Success requirements	4	
2	Des	ion	4	
_	2.1		4	
			4	
		· · · · · · · · · · · · · · · · · · ·	4	
	2.2		4	
	2.3		4	
	2.4	0	4	
	2.5	·	4	
	2.6	•	4	
	2.7		4	
		· ·	4	
		2.7.2 Test Plan	4	
3	Imr	lementation	4	
J	_		4	
4	Test	ing	4	
5	Eva	luation	4	

1 Analysis

1.1 Problem Identification

1.1.1 Problem Description

Popular inventory management solutions are relatively expensive, and may be out of reach for individuals or small schools. Inventory systems have numerous benefits for businesses and individuals alike; a business may choose to track their supply levels where an individual may wish to catelogue their DVD collection.

My goal is to create a web-based application aimed at both businesses and individuals to manage inventory, with additional modern features such as automatic item re-ordering when stocks are running low.

Traditional inventory management solutions are typically single-user at best, whereas I am to create a multi-user, collaborative environment.

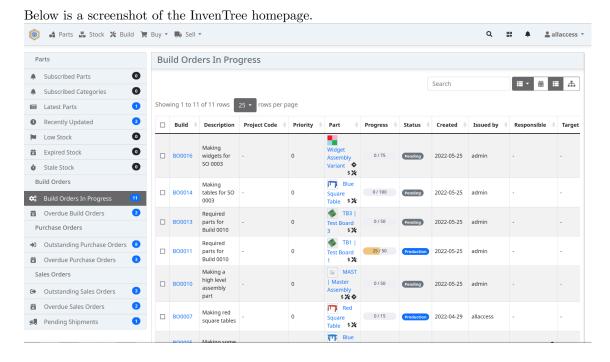
1.1.2 Interview

1.1.3 Existing similar solutions

InvenTree https://inventree.org/

Overview

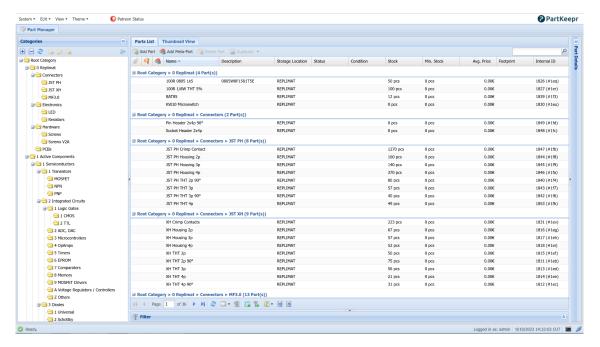
InvenTree is an **open-source** inventory management system, providing *low level stock control and part tracking*. It uses a Python/Django database backend and provides both a **web-based interface** as well as a REST API for interacting with other services. InvenTree also has a powerful plugin system for custom applications and other extensions.



Parts applicable to my solution

- concept is similar (web-based), but I'm doing a different approach.
- not indented for stock control

${\bf PartKeepr\ https://partkeepr.org/}$



Overview

PartKeepr is an open-source inventory management system with a focus on electronic components. It is designed around four main principles:

- Fast Part Searching
- Ability to add complete part database
- Keeping track of stock
- Ease of use

- 1.1.4 Features to be incorporated into solution
- 1.1.5 Feedback from stakeholders
- 1.2 Requirements
- 1.2.1 Stakeholder requirements
- 1.2.2 Software and hardware requirements
- 1.2.3 Success requirements

2 Design

- 2.1 User Interface Design
- 2.1.1 Usability Features
- 2.1.2 Feedback from stakeholder
- 2.2 Modular breakdown
- 2.3 Algorithms
- 2.4 Data Dictionary
- 2.5 Inputs and outputs
- 2.6 Validation
- 2.7 Testing
- 2.7.1 Methods
- 2.7.2 Test Plan
- 3 Implementation
- 3.1 First Iteration
- 4 Testing
- 5 Evaluation