



DELIVERING VALUE THROUGH INNOVATION – EVERY PROJECT, EVERY SOLUTION, EVERY TIME.

# **Project Tree**

## **A Project Management System**

### **SU Software Engineering Project Proposal**

# Introduction

The project entails the development of a project management system. The system represents a project visually as a tree structure, with each “leaf” being a project task, and each branch being a task dependency. The core idea of the system is to allow for all task dependencies to be captured accurately and clearly, to ensure effective project management.

## Background and Motivation

This system represents an alternative approach to traditional project management. The clear and accurate representation of task interdependencies will allow for better project control and management.

Working on this project will develop various aspects of software development, including database structures and interdependencies (as well as other server-side technologies). The system requires an interactive graphical interface that allows users to design project plans easily and quickly. This aspect will promote the development of front-end development skills.

The system can be applied to actual projects, allowing companies to achieve project execution on budget and schedule.

## System (Functional) Requirements

### System Core Requirements

1. General project plan layout and configuration
  - i. A web app is required that allows users to build a project plan (schedule) in a tree format using a graphical user interface. The project tree must be based on a database back-end.
  - ii. All project tasks must be represented by nodes that can be fully configured with all task information:
    - a. Task name
    - b. Task ID (unique, auto generated)
    - c. Responsible person (responsible for task)
    - d. Package manager (responsible for group of tasks)
    - e. Start date
    - f. Duration
    - g. End date
    - h. Task resources (people)
    - i. Task progress

- iii. Task interdependencies (connections) must link task objects (nodes) and transmit task dependencies (specifically child start date) between tasks.
  - iv. The system must be able to create views of tasks to allow for interdependencies to be drawn without connector lines crossing and becoming disorganized.
2. Project metrics and details. The system must be able to:
  - i. Display the project critical path. Critical path can be determined by the shortest and/or the longest path to the final node.
  - ii. Display (highlight) all task assigned to specific resources.
3. Project save and load
  - i. The system must provide for project trees to be saved and loaded. The format that this is done can be chosen by the students.

## **System Optional Requirements**

1. Conversion of project tree to Gantt chart.
2. Project progress graphical dashboard.
3. Export project tree to Excel and pdf.

## Architectural Requirements

At least 3 items (e.g. Scalability, Security, Performance).

## Project Management

1. Use version control tool for all code. E.g. Github, Bitbucket, Gitlab
2. Use a project management tool to track progress. E.g. Trello, Asana
3. Use a testing and continuous integration framework of your choice.

## Technology Specifications

- React Native
- Node.js
- Heroku web server
- Neo4j

## System Design

A prototype of the system has been developed in Excel VBA (will be provided to the students). This will serve as the basis of the design. Most of the core functionality is included in this prototype.

Core components of the system will comprise of:

- Application database
- Web application

The following images give an overview of the system (from prototype).

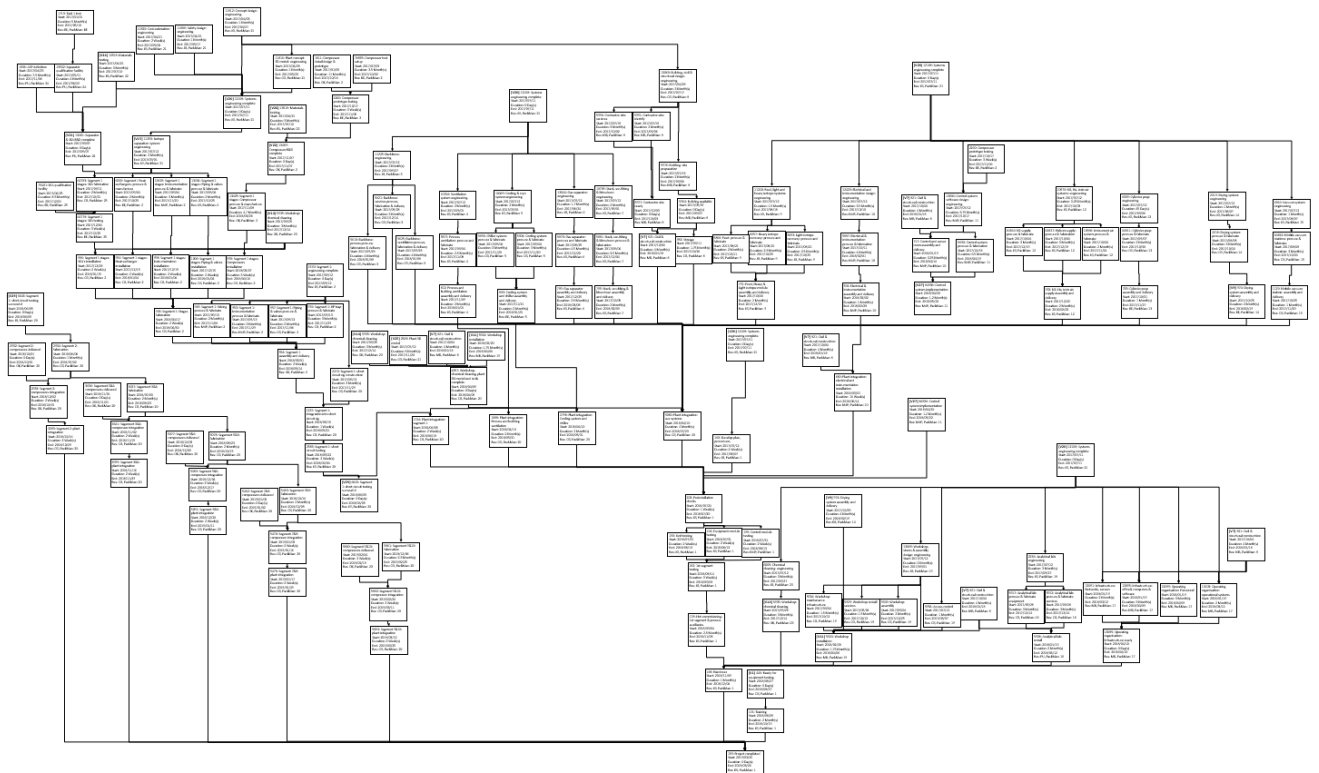


Figure 1: Sample project tree

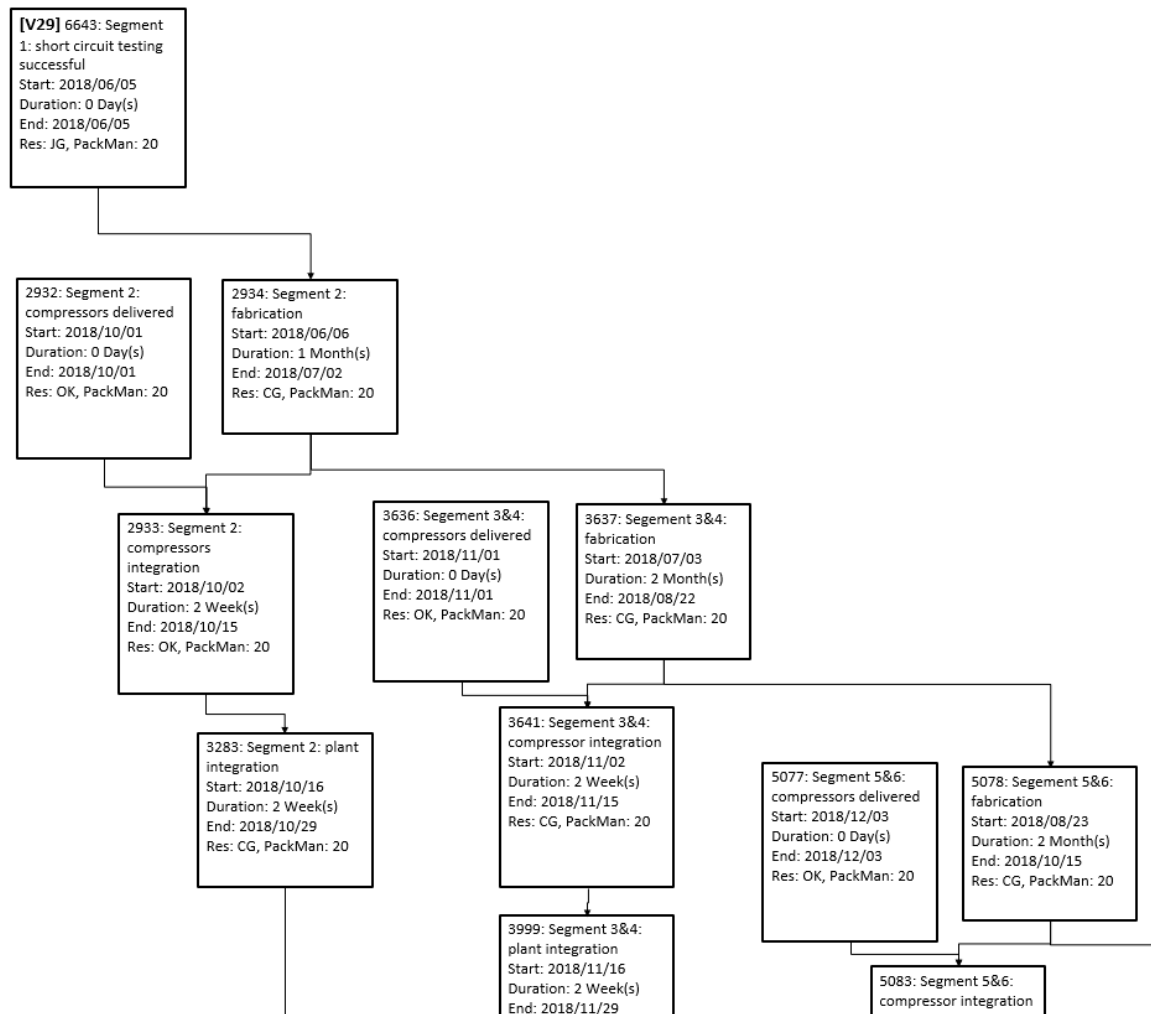


Figure 2: Detail view of tasks in tree

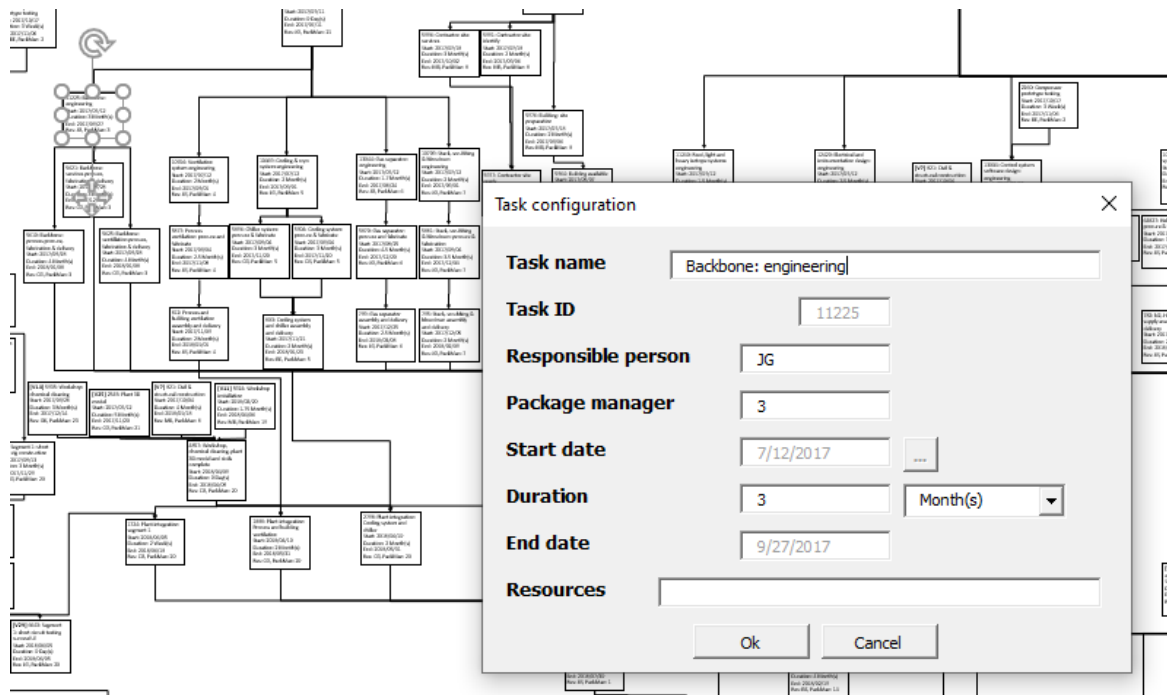


Figure 3: Task configuration interface

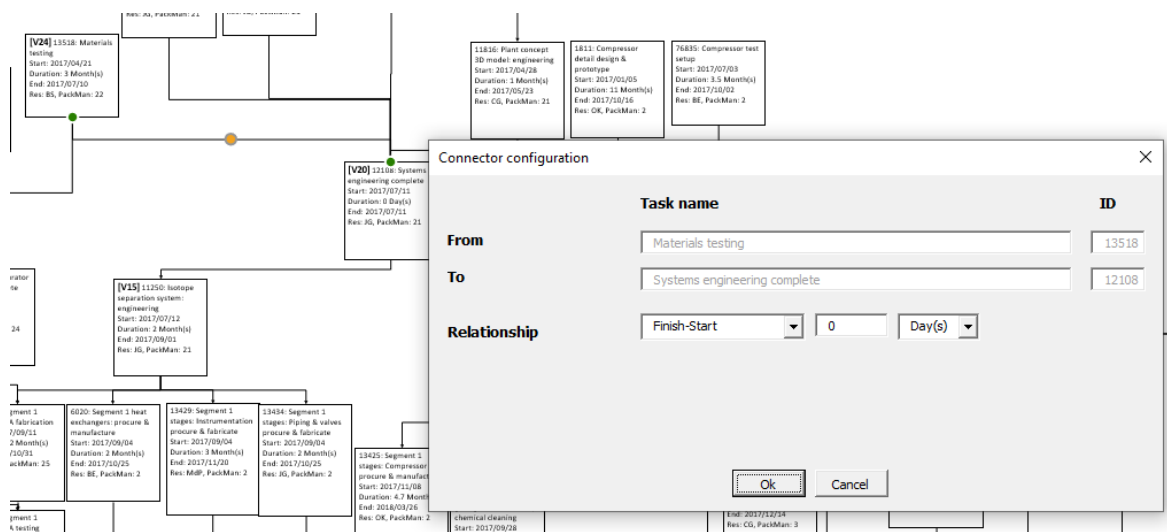


Figure 4: Dependency (connector) configuration interface



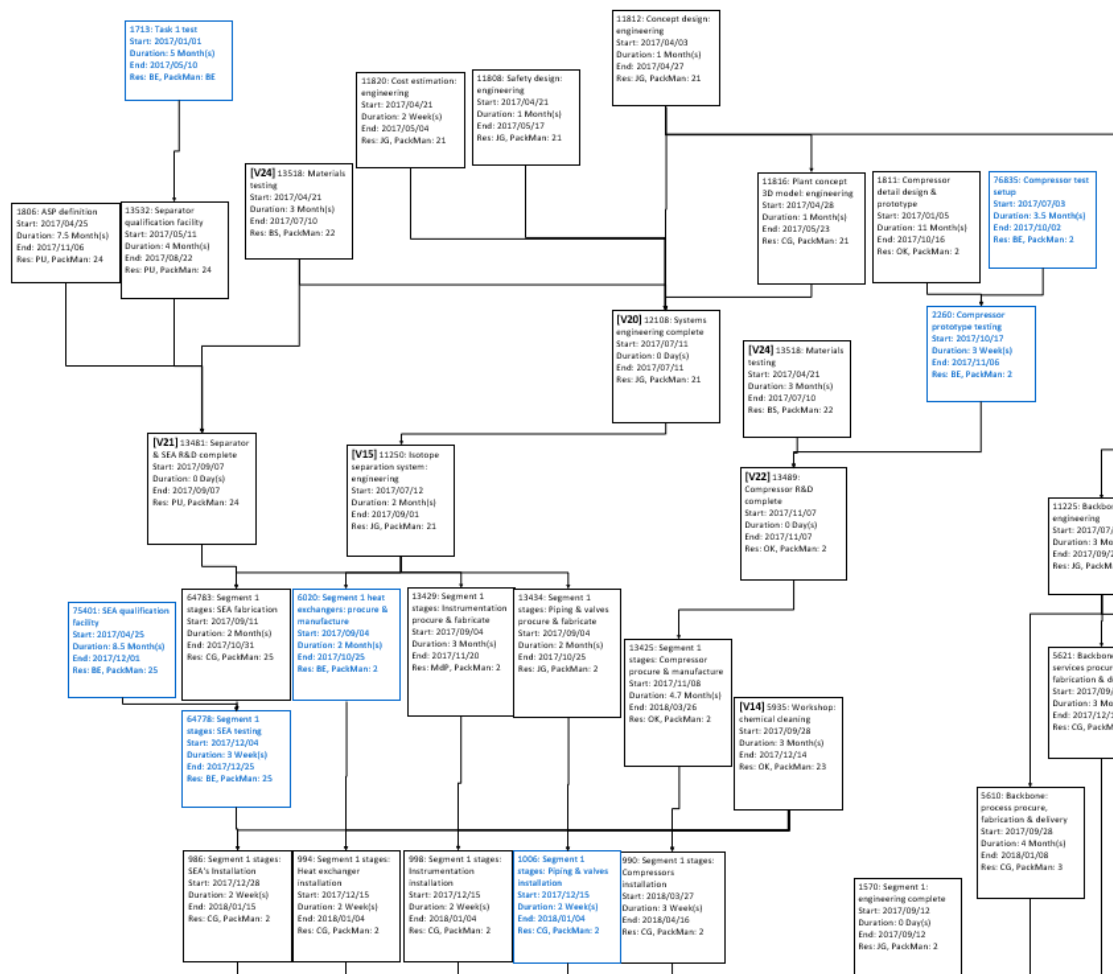


Figure 6: Tasks assigned to specific person indicted in blue.