CS455A: Software Engineering

Ed Eater Estimate

Submitted By
Harshit Raj (200433) | Rashmi G R (200772)
Akanksha Singh (200076) | Abhimanyu Sethia (190023)
On Sunday, 10 September 2023

Requirements

Segment	High Priority	Medium Priority	Low Priority
Baseline	Get an HTML editor with simultaneous rendering		
UI/UX	Design and implementation		
File IO	• Open • Close • Save	• Exporting to other formats	• Tabs
Editing	• Underlining • Bold • Italics • Colors • Fonts • Size	• Link • Case-Setting • Spacing • Alignment • Bullets/Numbering (non-nested) • Alignment • Blockquotes/Codeblocks	• Bullets/Numbering (nested) • Strikethrough • Tables • Columns • Highlight • Image Insert • Equations • Margins
Additional Features	• Copy/Paste • Key Binding	• Version History • Themes	• Comments • Compare

Github Links

- Project Gantt board: https://github.com/users/1-Harshit/projects/2/views/1
- Repository: https://github.com/1-Harshit/ed-eater
- Issues: https://github.com/1-Harshit/ed-eater/issues

Estimated Time

- Boilerplate + Baseline App: 25 hrs
- Editing
  - High: 15 hrs
  - Medium: 25 hrs
  - Low: 25 hrs
- Shortcuts and additional features
  - High: 10 hrs
  - Mid-low: 30 hrs
- File IO + Exports
  - High: 15 hrs
  - Mid-low: 10 hrs
- UI/UX:
  - Design   20 hrs
  - Implementation   20 hrs
- Deployment
  - CI/CD Setup   5 hrs
- Alpha Testing   15 hrs

Timeline

Tasks	Week 1 (11-15)	Week 2 (18-22) *Midsem*	Week 3 (25-29)	Week 4 (2oct-6)	Week 5 (9-13)	Week 6 (16-20)
Baseline	X					
UI/UX	X	X			X	X
Editor Features			X	X		
File IO			X	X	X	
Shortcuts/ other features				X	X	
Deployment					X	X
Testing					X	X

Prioritized tasks

• Requirements and estimate (Completed) • Tech Stack (Completed)
• Design (UI/UX)
• Baseline Application
• Editor features (High Priority) • File IO (open/close)
• Editor features (Medium Priority) • File IO (Medium Priority) • Shortcuts/keybindings
• Editor features (Low Priority, if time permits) • File IO (Low Priority, If time permits) • Other features
• Deployment • Final UI/UX
• Alpha testing

Reasoning/Logic of Prioritization and Estimates

We began by extracting potential features of a text editor and classifying them into low-priority (luxury), mid-priority, and high-priority (essential) categories. Subsequently, we evaluated the feasibility of each feature from a developer's perspective, adjusting the priority matrix accordingly. The priority matrix was initially organized into sub-categories such as File IO and UI/UX. Following this, we constructed a dependency graph to determine task interdependencies. We identified tasks that could be executed in parallel and, taking into account both feasibility and task importance, formulated a Gantt chart for the project. Finally, we compiled a prioritized task list based on this assessment.