
Project Governance Document

Project: Round Robin Scheduler

Group: Abiola, Ochihai

Date Started: January 23, 2025

Code and Documentation Standards

Programming Languages:

- Primary: C++
- Secondary: Haskell

Naming Conventions:

C++:

- Variables: snake_case (e.g., match_length)
- Functions: snake_case (e.g., calculate_time())
- Classes/Structs: PascalCase (e.g., Tournament)
- Constants: UPPER_SNAKE_CASE (e.g., MAX_PARTICIPANTS)
- Files: snake_case (e.g., read_input.cpp)

Haskell:

- Variables: camelCase (e.g., matchLength)
- Functions: camelCase (e.g., calculateTime)
- Types/Modules: PascalCase (e.g., Tournament)
- Constants: camelCase (e.g., maxParticipants)
- Files: PascalCase (e.g., ReadInput.hs)

File Structure:

- All source files must be organized in logical modules
- Header/source separation for C++ (.h/.cpp)
- One module per file in Haskell
- Test cases in separate /test directory

Comments:

Function Documentation:

- Brief description of purpose
- Parameter descriptions
- Return value explanation
- Pre/Post conditions where applicable

File Headers:

- File name
- Author(s)
- Creation date
- Last modified date
- Brief description of purpose

Inline Comments:

- Explain complex logic
- Note non-obvious optimizations
- Mark temporary code with TODO/FIXME

Error Handling:

- Validate all user input
- Check file operations (open/read/write)
- Use descriptive error messages
- Handle edge cases explicitly
- Log errors with context

Coding Guidelines:

Function Rules:

- Maximum 40 lines per function
- Single responsibility principle
- Clear input/output contracts

File Documentation:

- Each source file must include a header with the file name, author(s), date, and purpose.

Inline Comments:

- Use inline comments where appropriate to clarify potentially confusing sections of code.

Development Timeline

Task	Team Members	Completion Date
Concept Memo	Abiola, Ochihai	January 24
Concept Memo Redo	Abiola, Ochihai	January 30
Create Modules, Makefile, and Header Files	Abiola	February 13
Data Structure Module	Abiola	February 16
Read Input Module	Abiola	February 16
Main Module	Abiola, Ochihai	February 20
Print Output Module (tested with hardcoded instances)	Abiola	February 20
Scheduler (Solver) Module	Ochihai, Abiola	March 1
Test Cases	Abiola	March 6

Documentation	Ochihai	March 6
Documentation	Ochihai	March 7
Command Line Arguments	Abiola	March 23
Improved Scheduler	Abiola	March 26
Test Cases	Ochihai	March 26
Haskell Structures Completion	Abiola	March 27
Haskell Input, Output, Main, Scheduler	Abiola, Ochihai	April 1
README/Project Functionality Document	Abiola	April 2
Test Plan Document	Ochihai	April 4
Final Haskell Documentation Completion	Ochihai	April 5
Final C++ Documentation Completion	Abiola	April 5
Project Governance Document	Ochihai, Abiola	April 6
Final Test Comparison between both Solutions	Abiola	April 6

Group Meetings and Development Activity

Communication/Meeting Locations

- In-person meetings: computer lab beside math help room
- Online Meetings: Discord
- Communication: iMessage
- Version Control: Github

Meeting Minutes

Meeting Date: January 23

- Agenda: Decide Concept for Programming Project
- Actions:
    - Brainstormed ideas
    - Discussed communication and division of labour

Meeting Date: January 28

- Agenda: Redo Concept Memo
- Actions:
    - Received feedback on how to improve the concept
    - Brainstormed new ideas
    - Fixed original concept memo

Meeting Date: February 4

- Agenda: Determine Project Data Structure
- Actions:
    - Discussed possible data structures for the scheduler
    - Drew out data structure on a whiteboard

Meeting Date: February 20

- Agenda: Worked on Modules
- Actions:
    - Met on Discord to finish to focus on output and solver
    - Changed data structure from 2D array to 1D array

Meeting Date: March 4

- Agenda: Peer Review

- Actions:
    - Got feedback from another group (Ravi and Jayden)
        - Project Governance Document
        - Write output to file instead of onto the console

Meeting Date: March 6

- Agenda: Finalize Imperative Solution
- Actions:
    - Reviewed the full implementation
    - Debugged and tested the final solution
    - Prepared for interim project submission
    - Clean up governance document

Meeting Date: March 7

- Agenda: Finalized Imperative Solution
- Actions
    - Reviewed the full implementation
    - Added Inline Comments

Meeting Date: March 23

- Agenda: Finalized Imperative Solution
- Actions:
    - Reviewed the full implementation
    - Added command line arguments

Meeting Date: March 26

- Agenda: Finalized Imperative Solution
- Actions:
    - Improved scheduler
    - Added test cases

Meeting Date: March 27

- Agenda: Begin Haskell Implementation
- Actions:
    - Haskell Data Structures Completed

Meeting Date: April 1

- Agenda: Continuing Haskell Implementation
- Actions:
    - Input, Output, Main and Scheduler Modules Completed

Meeting Date: April 2

- Agenda: Continuing Haskell Implementation
- Actions:
    - Adding more test cases/test plan

Meeting Date: April 4

- Agenda: Continuing Haskell Implementation
- Actions:
    - Discussed what needs to be done
    - Updated readme
    - Updated project governance

Meeting Date: April 5

- Agenda: Finishing Off Overall Project
- Actions:
    - Updated c++ documentation
    - Added Haskell documentation
    - Adding more test cases/update test plan

Meeting Date: April 6

- Agenda: Finishing Off Overall Project
- Actions:
    - Fixing test cases
    - Finishing off project governance