

RESPONSIBILITIES

Phase 5

- Ensure Skeleton Classes are complete
- **Create a software test plan**
 - Document proposed test cases
 - Plan out project test suite
 - Begin coding junit tests (**have at least one or more of a basic test using junit ready by due date**)
- Begin planning and creating prototypes
-
- Deliverables:
 - Project Status Report
 - Individual Git Log
 - Link to GitHub Repo
 - Names of team members

Phase 5 Roles:

- **Yew:**
 - Check class UML document: does it make sense? Update if necessary. Complete by end of day Thursday.
 - **Avin:**
 - Check Requirements document: is it complete? Update if necessary. Complete by end of day Thursday.
 - updated player.java
 - **Ivan:**
 - Start a junit testing doc: complete rough version by Thursday end of day; get feedback
 - update gantt chart with - junit tests, junit docs, junit test suite class
 - update roles doc
 - **For grabs:**
 - Ensure Skeleton Classes are complete [rough draft is OK]
 - Build Test Suite [rough version, just a starter].
 - Test that all constructors are not null.
 - Test all other available functions work as intended.
 - Come up with test cases which could break the program.
 - Begin planning & creating prototypes
 - Offline GUI or Command line version of the game
-

GENERAL ROLES

Ivan - Front End / GUI / UI work / simulated objects / general code editing // overall project management (making sure deadlines are being hit, updating the roadmap & schedule)

Avin - Classes / Objects (making sure all the object content is correct) / assistant project manager (helping make sure the schedule and roadmap is correct and feasible) / Requirements document

Yew - Database / assistance with front end and back end -> all-around code management

Original Document:

Roles and Responsibilities

JianTing Tan – create the basic objects of the Card and Deck. Implements methods associated to them.

Zhenwen Wang – create the basic objects of the Player and Playersize (gamesize). Implements method associated to the classes.

Isabell Ampon – create the basic Money objects. Implements method associated with the class.

All – Work together to implement the authentication, rules, and interactions of the game.