1. Project Team Name and #. List of team members. Vision. Project description.

Project Team Name: GameDashBoard 08
Team Members:
Yang Yang
Zhaozhong Peng
Yuhou Wang

Vision: Create a place where you can discuss games and Compete against one another in tournaments

Project Description:
Originally, we wanted to implement both a game dashboard and a tournament hosting site

We only implemented tournament hosting where users can create tournaments, sign up for tournaments, and compete against one another after being matched by our system. The user can also delete tournament, delete team, and report score/match results.

2. List the features that were implemented (table with ID and title).

Implemented User Requirements:

UR-018	as a user/admin, I should be able to register for a tournament	Join Tournament	User, Admin	medium
UR-009	As a admin, I should be able to create tournaments.	Host Tournament	Admin	High
UR-019(NEW	As a User, I should be able to host a tournament	Host Tournament	User	High
UR-020(NEW	As a User, I should be able to delete a tournament	Delete Tourn	User	High
UR-021(NEW	As a User, I should be able to report my match score	Report Score	User	High
UR-022(NEW	As a User, I should be able to unregister from tournament	Unregister Tourn	User	High
UR0-23(NEW	As a User, I should be able to view my opponent	View Opponent	User	High

3. List the features were not implemented from Part 2 (table with ID and title).

User Requirements				
ID	**Requirement**	**Topic Area**	**Actor**	**Priority**
UR-001	As a admin/User, I should be able to create posts.	Create Post	User, Admin	High
UR-002	As a admin/User, I should be able to edit posts.	Edit Post	User, Admin	High
UR-003	As a admin/User, I should be able to delete posts.	Delete Post	User, Admin	High
UR-004	As a admin/User, I should be able to upvote posts.	Upvote	User, Admin	High
UR-005	As a admin/User, I should be able to downvote posts.	Upvote	User, Admin	High
UR-006	As a admin, I should be able to ban users.	Ban user	Admin	medium
UR-007	As a admin, I should be able to host tournaments.	Host Tournament	Admin	High
UR-008	As a user, I should be able to send the request of creating tournaments to admin	Host Tournament	User	High
UR-010	As a user, I should be able to join tournaments.	Join Tournament	User	High
UR-011	As a admin, I should be able to edit tournaments.	Host Tournament	Admin	High
UR-012	As a user/admin, I should be able to Search posts.	Search Post	User, Admin	High
UR-013	A new viewer should be able to Sign Up	Sign Up	User	High
UR-014	As a user/admin, I should be able to Login	Login	User, Admin	High
UR-015	As a user/admin, I should be able to Logout	Login	User, Admin	High
UR-016	As an admin, I should be able to create new game subreddits	Create Subreddit	admin	High
UR-017	as a user/admin, I should be able to select the game subreddit I want to browse	Control Subreddit	user, admin	High

Business Requirements				
ID	**Requirement**	**Topic Area**	**Actor**	**Priority**
BR-001	A User cannot have more than 1 concurrent pending tournament form	Join Tournament	User	High
BR-002	A User cannot be registered to two tournaments simultaneously	Join Tournament	User	High
BR-003	A User cannot make more than 1 post per 30 seconds.	Create Post	User	High
BR-004	A User cannot make posts with empty content.	Create Post	User	High
BR-005	A username must contain at least 4 characters and at most 10 characters.	Authentication	Admin, Users	High
BR-006	A password must contain at least one capital letter and one number.	Authentication	Admin, Users	High
BR-007	All users must have e-mail addresses.	User Account	User	High

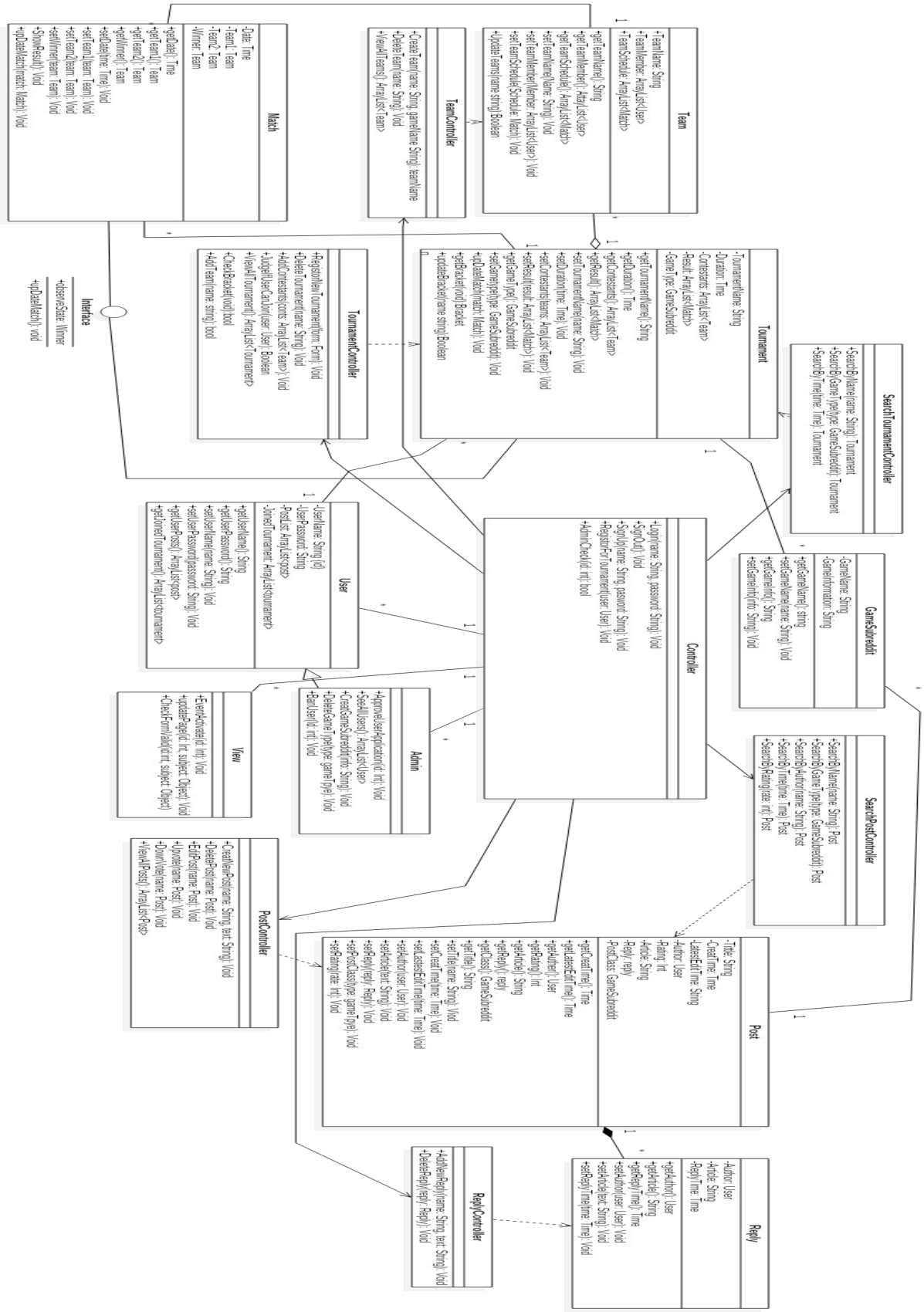
Functional Requirements				
ID	**Requirement**	**Topic Area**	**Actor**	**Priority**
FR-001	As a system, a list of users should be displayed for admin to ban users.	Ban User	Admin, system	Medium
FR-002	As a system, a list of posts should be displayed in the home page.	Create Post	Admin, user, system	High
FR-003	As a system, posts should be sorted by votes in the home page.	Upvote	User, system	Medium
FR-004	As a system, the results of searches should be listed by name.	Search Post	User, system	Medium
FR-005	As a system, the process of tournaments should be renewed on time.	Host tournament	Admin, user, system	Medium
FR-006	As a system, it should be display the list of tournament information when user click tournament pages.	Join tournament	User, system	High

FR-007	As a system, it should be update brackets and notify players on time.	Join tournament	User, system	Medium
FR-008	As a system, it should display tournament form which user need to fill out.	Host tournament	User, system	Medium
FR-009	As a system, it should check the form from user and send to admin when user already filled in all requirements.	Host tournament	User, system	Medium
FR-010	As a system, it should register a new signed up User to its database	Sign Up	User, system	High
FR-011	As a system, it should verify the login of a user/admin	Login	User, admin, system	High
FR-012	As a system, it should update the status when a user/admin logsout	Login	User, admin, system	High
FR-013	When a new subreddit is created, it is processed and shown on the site	Create Subreddit	system	Medium
FR-014	When a user joins a tournament, the bracket is updated with the userinfo	Join tournament	User, system	Medium

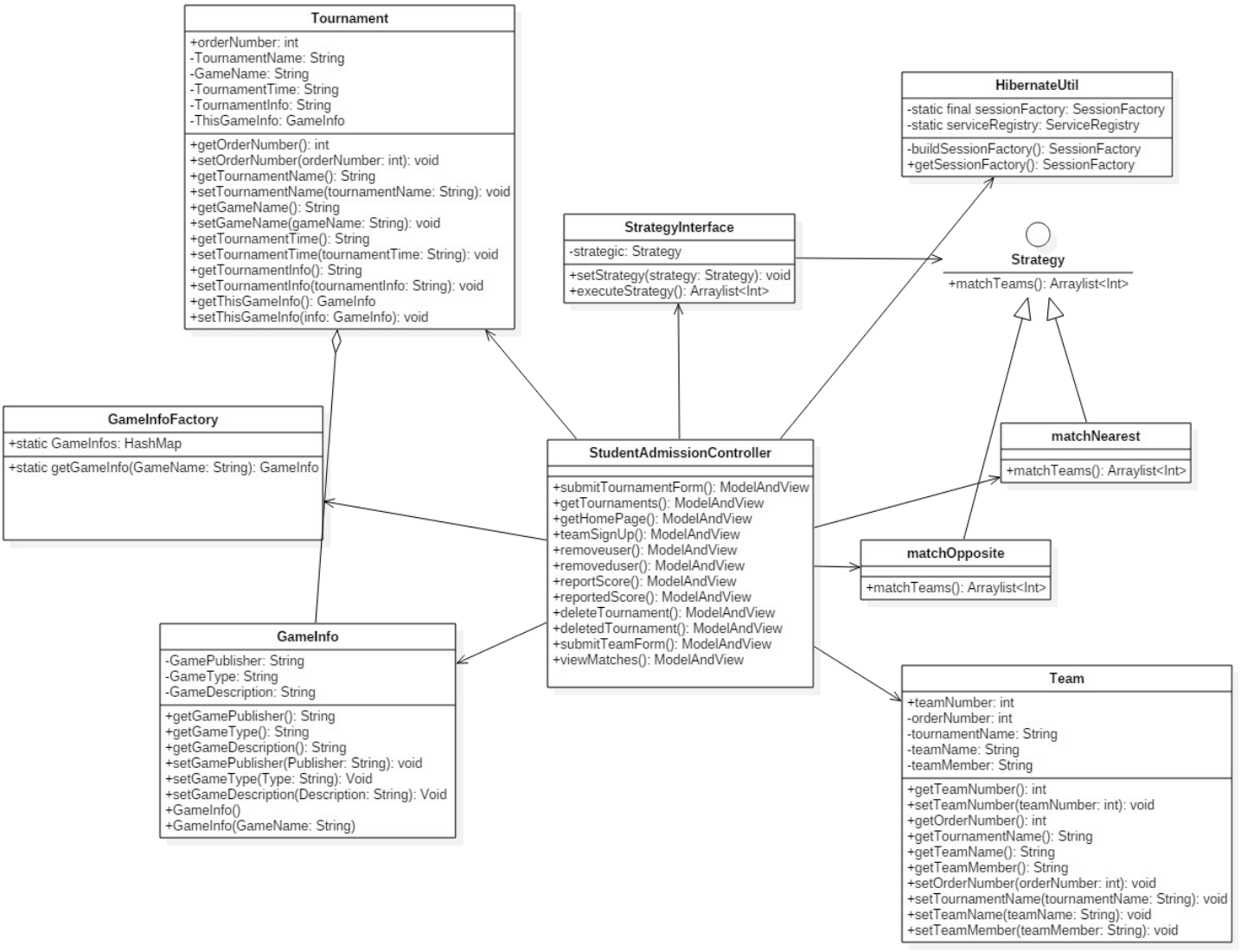
Non-Functional Requirements		
ID	**Requirement**	**Priority**
NFR-001	**Reliability** - The entires of all data must store with a reliable way.	High
NFR-002	**Security -** All data must tranfer with database in secure way.	High
NFR-003	**Performance** - All posts can be displayed in 12 seconds and can be searched in 3 seconds.	medium
NFR-004	**Platform constraints** - all functionalities should be used in all website browsers.	low

4. Show your Part 2 class diagram and your final class diagram.

Part 2 Class Diagram:
https://github.com/niru32868/CSCI4448Project/blob/master/Part2ClassDiagram.png



Final Class Diagram:
https://github.com/niru32868/CSCI4448Project/blob/master/FinalClassDiagram1.png

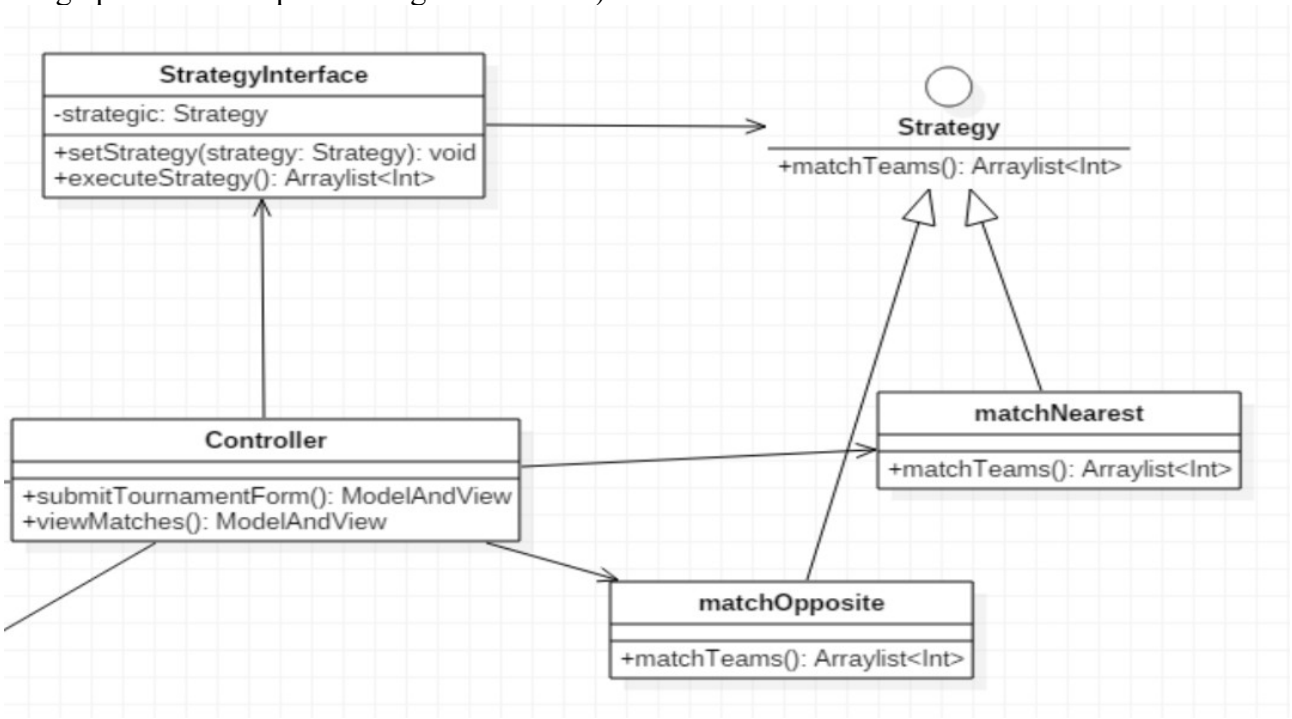


What changed? Why? If it did not change much, then discuss how doing the design up front helped in the development.

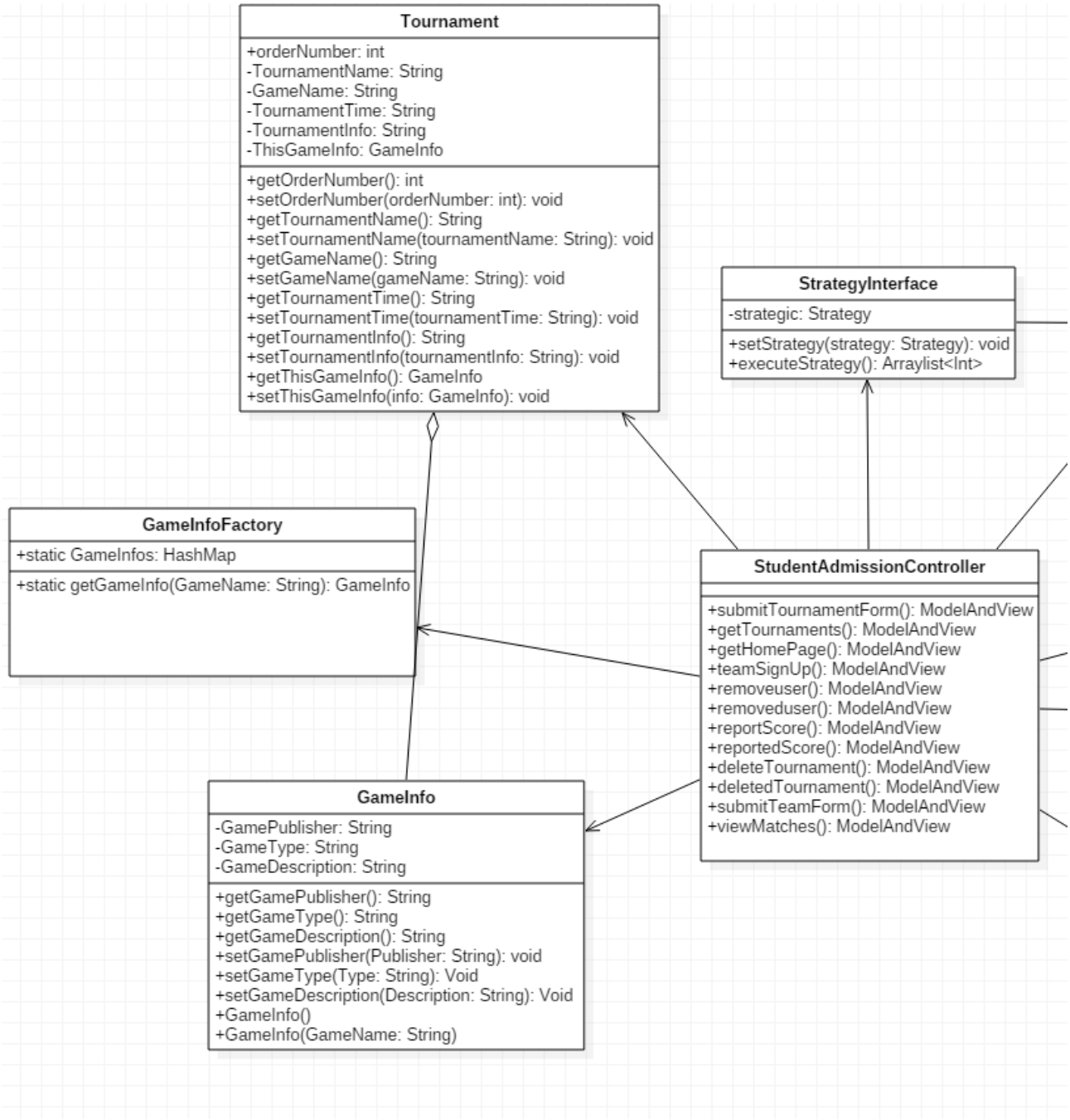
A lot has changed. For instance, we did not implement the Discussion Board, and only implemented the tournament section. In addition, we did not implement user/admin, but only has user which did not require a login. There are also the addition of Design Patterns like strategy and flyweight!

Also, view and frontcontroller are not classes, so those were not included. We also did not seperate the controller into seperate pieces, like a tournamentcontroller, teamcontroller, etc. We did not implement as much as we originally planned, which was too ambitious.

5. Show the classes from your class diagram that implement each design pattern (each design pattern as a separate image in the .PDF).



The strategy algorithm is chosen at runtime when viewMatches is actived. The two classes that implements Strategy are matchOpposite and matchNearest. StrategyInterface class is used to program to an interface not an implementation.



Since only three games can be seletcted to be made tournaments for. We can share the GameInfo class as flyweight even as the number of Tournaments increase to 50 and beyond. Gameinfo is an aggregate of Tournament.

6. What have you learned about the process of analysis and design now that you have stepped through the process to create, design and implement a system?

The biggest lesson I have learnt is that it is extremely important to know the framework you are working with when designing the system.

This extentiates to everything. Even in the case of use cases/desired functionalities, it is necessary to know the framework and how difficult it will be to implement the functionality and only then if it is feasible to implement it. Since the beginning of the project, our scope of project was beyond our means.

Similarly, in the case of class/sequence diagrams, our mapping was completely wrong.

The second lesson I have learnt is to be agile in my approach. This means changing planned functionalities as time goes on in accordance to the team's abilities. Even if the framework is known, the estimated amount of necessary effort can still be off. Thus, it is important to be agile in one's approach to design and implementation of a system.

The third lesson I have learnt is to leave time for refactoring! There are a lot of refactoring/improvement that can be made to the code that does not involve adding new functionality. However, out of time, so no refactoring can be done. If the project were to be opened up at a later date and used, it would be much harder to refactor the code, whether it by the team or someone else entirely. Thus, I think leaving time to refactor the code is extremely important.

The fourth lesson I have learnt is to aim low and then climb high, rather than aim high and go low. Aiming low and achieving high boosts one's morale and would look better to the customer in terms of results. However, there should be a balance as so to not cheat the customer!