Journal for Games

IA Project (Phase One)

Problem Definition:(Enter Six paragraphs for the problem definition, as you can see we have already added one by default, so each one of the team have to brainstorm and write one problem definition, there are five paragraphs left for you)

1. Users that interested in Games and want to know all the latest information about it they have to reach all the Stores of Games in the market or open the various websites to get this information, which will cost the user a lot of money and waste of time.
2. User can view and review products.
3. User can buy product.
4. User can rank the product or add it to his favorite.
5. User can view products as :  
   - Top popular.  
   - Top paid.  
   - Top free.
6. Admin can add , delete and update game or category.
7. Admin can show any statistics he need about the whole system.
8. Admin can Block or unblock users.
9. Admin can also add another admin.

User Requirement: (We needfourteen paragraphs for the User requirement, as you can see we have already added nine paragraphs by default, so each one of the team have to brainstorm and write one User requirement, there are five paragraphs left for you)

1. We need a web application (Desktop Application) which demonstrates a system that integrates the entire offline GamesStores with the entire online Games (web sites) together.
2. We need a web application (Mobile Application) which demonstrates a system that integrates the entire offline Games Stores with the entire online Games (web sites) together.
3. The Web application includes two layout:
   1. First Layout: the User Layout and we will call it the Wall (all the Games news exists here).
   2. Second Layout: the Editor Layout and we will call it the Factory (news are conducted and edited here).
4. Editors are responsible for Writing and adding articles from the Factory (it is like an Editor profile) then after the acceptance of the admin, the article reflected to the Wall so the user could see it.
5. Admin could add articles from the Factory (profile), and then article reflected to the Wall so the user could see it (no acceptance needed).
6. An Admin that would be the chief editor that responsible for the acceptance and refuse of the entire articles added by editors on the Wall.
7. Editors, Admin and Users are authenticated to enter the Wall (no Login is needed).
8. Editors and Admin are authenticated to Enter the Factory (Login is needed).
9. Admins are authorized to edit the Wall.
10. Authorized users can view all games in the store and can also view other users review for specific game or category of games .
11. Authorized users After viewing specific Game can buy this game .
12. User can rank specific game by giving it score from 5 point.
13. User can add specific game to favorite list (recommend gamed to other players, game to played later) this favorite list helps user to reach his own list game in fast way.
14. User can view the products in the store by some criteria (filter ) as :  
    - Top popular (most downloaded game (most played)  
    - Top paid (games with highest cost).  
    - Top free.(top free games that played with user or recommended by other user or have high rank)
15. Admin can manage or has a full control of games stored on the system (he can view all games and control on it with smart way).
16. Admin can also control on the whole games categories (category is a unified unit that defines or describe the games which are under it) by creating new categories , updating its games inside or the description below and delete the category.
17. The system provides some options to the admin for collecting statistics about it.
    1. Count of users.
    2. Count of games.
    3. Count of categories.
    4. Most ranked games.
    5. Most bought games.
    6. Most downloaded games.
18. Admin can block specific user from buy or review game and also can remove this block .
19. Admin can also add another admin with the same role in the web application .

Use Cases: (DrawFiveUse Cases,So each one of the team have to draw one Use Case that include the Default user requirements (there was nine reqs.) we gave you and only the requirement that have been add by you )

(Use Case no 5)

Member no 4 Turn

(Nine requirements + Member requirement = 10 requirements)

(Use Case no 4)

Member no 3 Turn

(Nine requirements + Member requirement = 10 requirements)

(Use Case no 3)

Member no 2 Turn

(Nine requirements + Member requirement = 10 requirements)

(Use Case no 2)

Member no 1 Turn

(Nine requirements + Member requirement = 10 requirements)

(Use Case no 1)

Team Leader Turn

(Nine requirements + Team leader requirement = 10 requirements)

Activity Diagram: (DrawFiveactivity diagrams,So each one of the team have to draw one activity diagram that include the Default user requirements (there was nine reqs.) we gave you and only the requirement that have been add by you)

(Activity no 5)

Member no 4 Turn

(Nine requirements + Member requirement = 10 requirements)

(Activity no 4)

Member no 3 Turn

(Nine requirements + Member requirement = 10 requirements)

(Activity no 3)

Member no 2 Turn

(Nine requirements + Member requirement = 10 requirements)

(Activity no 2)

Member no 1 Turn

(Nine requirements + Member requirement = 10 requirements)

(Activity no 1)

Team Leader Turn

(Nine requirements + Team leader requirement = 10 requirements)

Sequence Diagram: (DrawFiveSequence diagrams,So each one of the team have to draw one sequence diagram that include the Default user requirements (there was nine reqs.) we gave you and only the requirement that have been add by you)

(Sequence no 5)

Member no 4 Turn

(Nine requirements + Member requirement = 10 requirements)

(Sequence no 4)

Member no 3 Turn

(Nine requirements + Member requirement = 10 requirements)

(Sequence no 3)

Member no 2 Turn

(Nine requirements + Member requirement = 10 requirements)

(Sequence no 2)

Member no 1 Turn

(Nine requirements + Member requirement = 10 requirements)

(Sequence no 1)

Team Leader Turn

(Nine requirements + Team leader requirement = 10 requirements)

ERD and Class Diagram: (Design your DB where you can store and manage Admins, Editors and user Data + Data added by Team (if there were any more actors added by the team))

(Class Diagram)

(ERD Diagram)

Implementation (only Designs: Layouts, Content Views are required)

* Start Page and Layouts:

Wall Page

Factory Page

Navi

Footer

Header

Navi

Footer

Header

Welcome to your Application

Start

Some info about the App



Content

Content

Start Page

* Content Views:

Login Form (Design needed)

Wall News (Dummy Data needed)

Admin Could Add, Edit, Update and Delete Editor (Design needed)

Admin Could Add, Edit, Update and Delete News (Design needed)

Team Member no 1 (Design needed)

Team Leader Idea (Design needed)

Editor Could Add, Edit, Update His News (Design needed)

Admin Could Accept or refuse edited News (Design needed)

Team Member no 2 (Design needed)

Team Member no 3 (Design needed)

Team Member no 4 (Design needed)

* Important Notes:

1. You have two layouts not one and a Start Page.
   1. Wall Page Layout.
   2. Factory Page Layout.
   3. Start Page.
2. According to the above Diagram in the Implementation Phase, Your designs will be exactly the same Format.
3. Don’t forget to Design all the contents needed to achieve all the user requirements.
4. You havetwolayouts, One Start page and Many Content Views.
5. All the Pages are Connected together by Using
   1. @HTML action link (inner link, Action Name, Controller name) search in Google if you don’t know what action link means.
6. Use @HTML.BodyRender () method in the Body of wall layout and Factory Layout.
7. Views Folder should include:
   1. Shared Folder contains (wallLayout.cshtml and factoryLayout.cshtml).
   2. startPage.cshtml
8. Your app is responsive.
9. Bootstrap is required (Grid System)
10. Create Folder for Styles.
11. Create Folder For Scripts.
12. Work Hard to Get Bonus.

Last thing this is a Games app then make it looks like a Games app

Good Luck and see you in the Next Phase