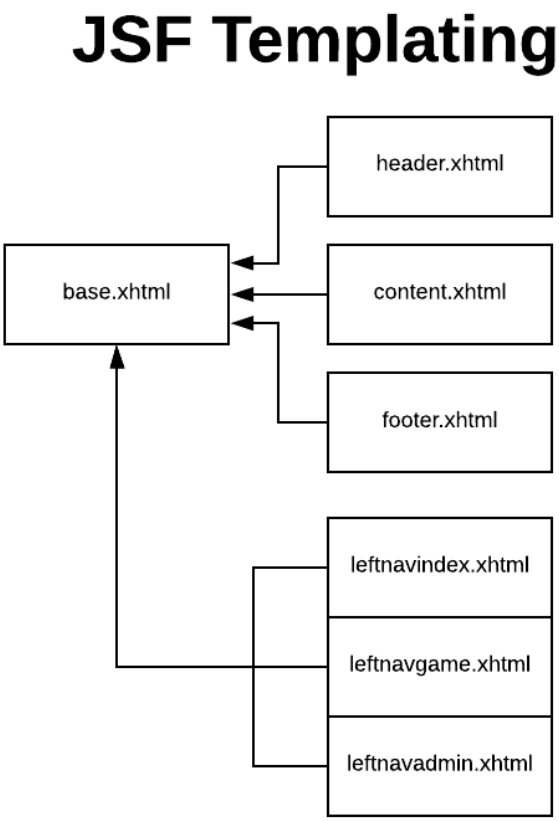
1. Requirements

F1. Any user can view an encyclopedia of unit and building information without logging in.
F2. Admin can suspend and unsuspend user accounts. Suspended players can log in, but cannot perform any actions and are notified that they are suspended.
F3. Admin can view, update, and delete user accounts.
F4. User can create an account with their own username, kingdom name, race, password, email. Email is used for security to send notifications to the player when sensitive actions are taken such as changing password or soft deleting their account.
F5. User can change their own password – need to enter current password to confirm.
F6. User can soft delete their account which suspends it. Only admin can unsuspend their account.
F7. Fiefdoms generate resources and can be attacked even though the user is not logged in
F8. An event log is generated of all the things that have happened since the player last checked their main page. This is displayed to the user when they check their main page.
F9. Player can explore to gain new land in order to build new buildings
F10. Player can spend gold and land to build different buildings that have different effects. For example, building towers helps them when they are attacked by another player. Building mines increases gold income per turn. Building farms increases peasant growth per turn.
F11. Player can spend gold and peasants to recruit different army units.
F12. Player can send diplomatic messages to other players, declare war, request peace, request alliance, and break alliances. Allies cannot declare war on each other. Players can only attack each other if they are at war.
F13. Players can reply to diplomatic messages, or accept/decline diplomatic requests (peace request, alliance requests).
F14. Player can send a chosen number of units from their army to attack an enemy player they are at war with. This generates a battle report which shows a log of the battle, who won, resources captured etc. If the attacking player wins, they capture some gold, land, and peasants from the defending player. Armies, buildings, resources are updated accordingly for both players.
F15. Battle reports are persisted and can be viewed at any time, but only by the players who were involved in the battle. Admin can also view battle reports.
F16. Players can search for and view basic information about all other player kingdoms. E.g. kingdom name, kingdom id, race, diplomatic status, rank, power.
F17. Each player kingdom has a power rating based on their army size, resources, buildings, land, which is updated every turn. Player rankings are updated accordingly in descending order. Player can view the ranking of all players, as well as their own ranking.
F18. Users can see how many other unique players are currently logged in
F19. Admin can create and remove news articles that can be viewed by anyone without having to be logged in. The most recent articles are displayed in a news feed on the index page.

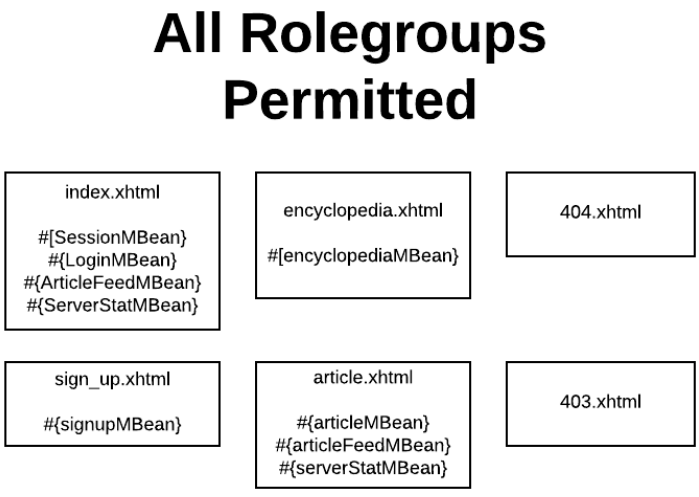
2. Software Design

3 tier architecture.

1. Web Layer



Glassfish resource security



Admin Rolegroup Only

admin.xhtml

suspend.xhtml
#{suspendUserMBean}

view_user.xhtml
#{viewUserMBean}

create_article.xhtml
#{createArticleMBean}

update_user.xhtml
#{updateUserMBean}

delete_article.xhtml
#{deleteArticleMBean}

delete_user.xhtml
#{deleteUserMBean}

User Rolegroup

account.xhtml
#{accountMBean}

create_message.xhtml
#{composeMBean}

rankings.xhtml
#{rankingsMBean}

battle_reports.xhtml
#{searchBRMBean}

diplomacy.xhtml
#{inboxMBean}
#{relationsMBean}

recruit_army.xhtml
#{recruitMBean}

buildings.xhtml
#{buildingsMBean}

explore.xhtml
#{buildingsMBean}
#{exploreMBean}

repport.xhtml
#{battleReportMBean}

change_pw.xhtml
#{changePassMBean}

message.xhtml
#{viewThreadMBean}

search_fiefdom.xhtml
#{searchFiefdomMBean}

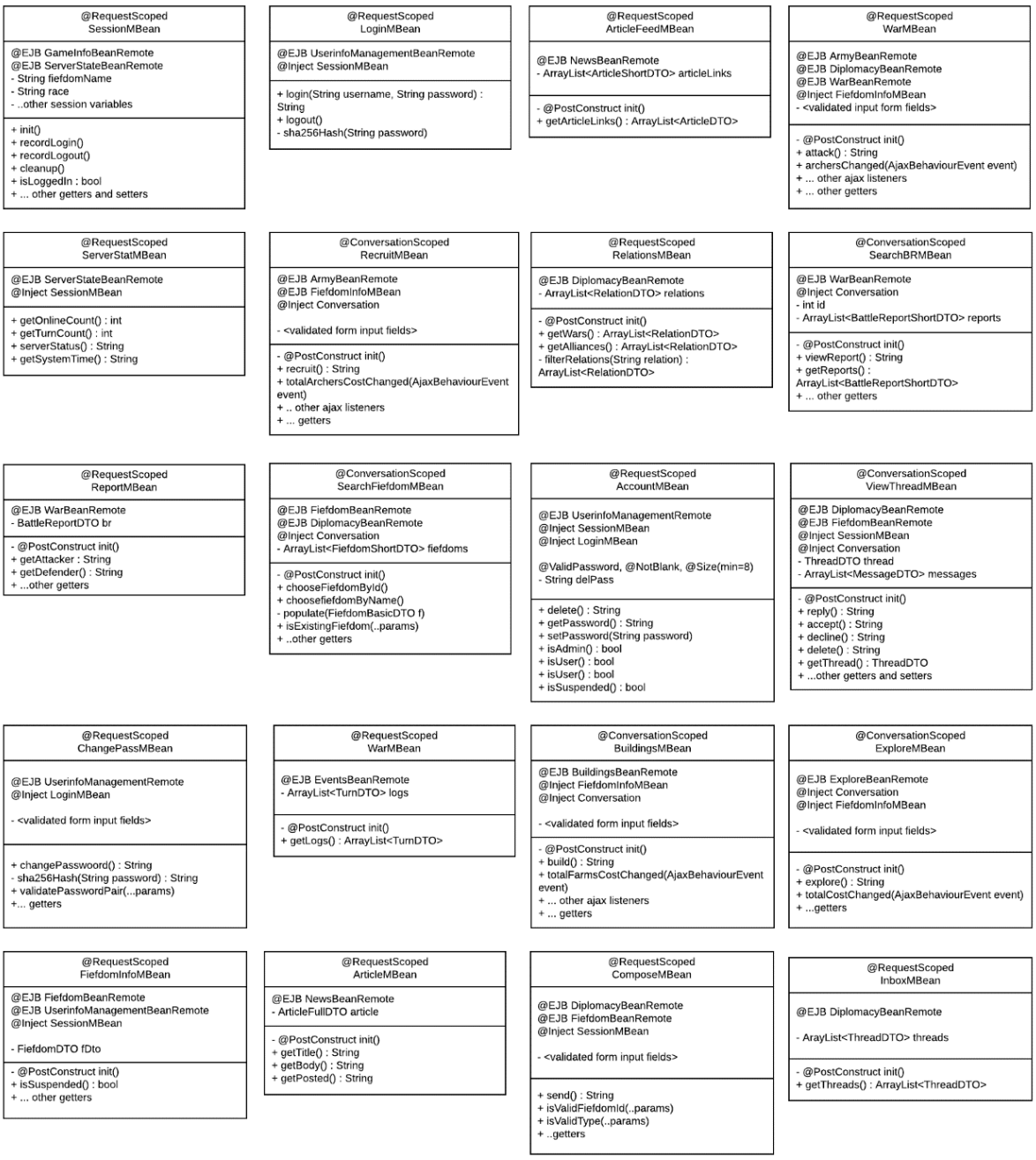
sign_up.xhtml
#{sessionMBean}
#{fiefdomInfoMBean}

war.xhtml
#{warMBean}

User and Suspended Rolegroups

main.xhtml
#{accountMBean}
#{loginMBean}
#{fiefdomInfoMBean}
#{serverStatMBean}
#{sessionMBean}
#{eventLogMBean}

Web Layer
Backing Beans



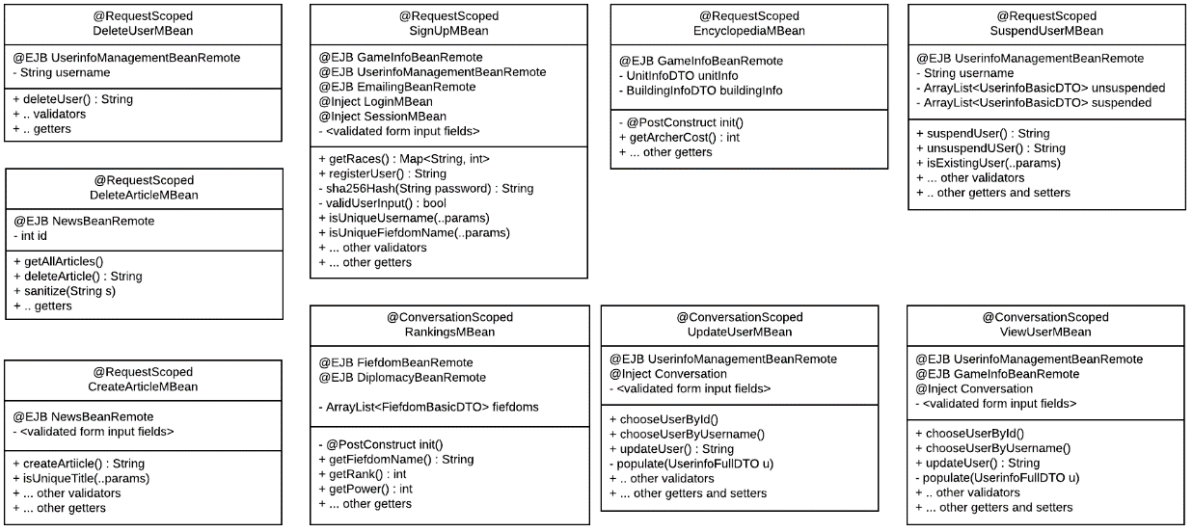
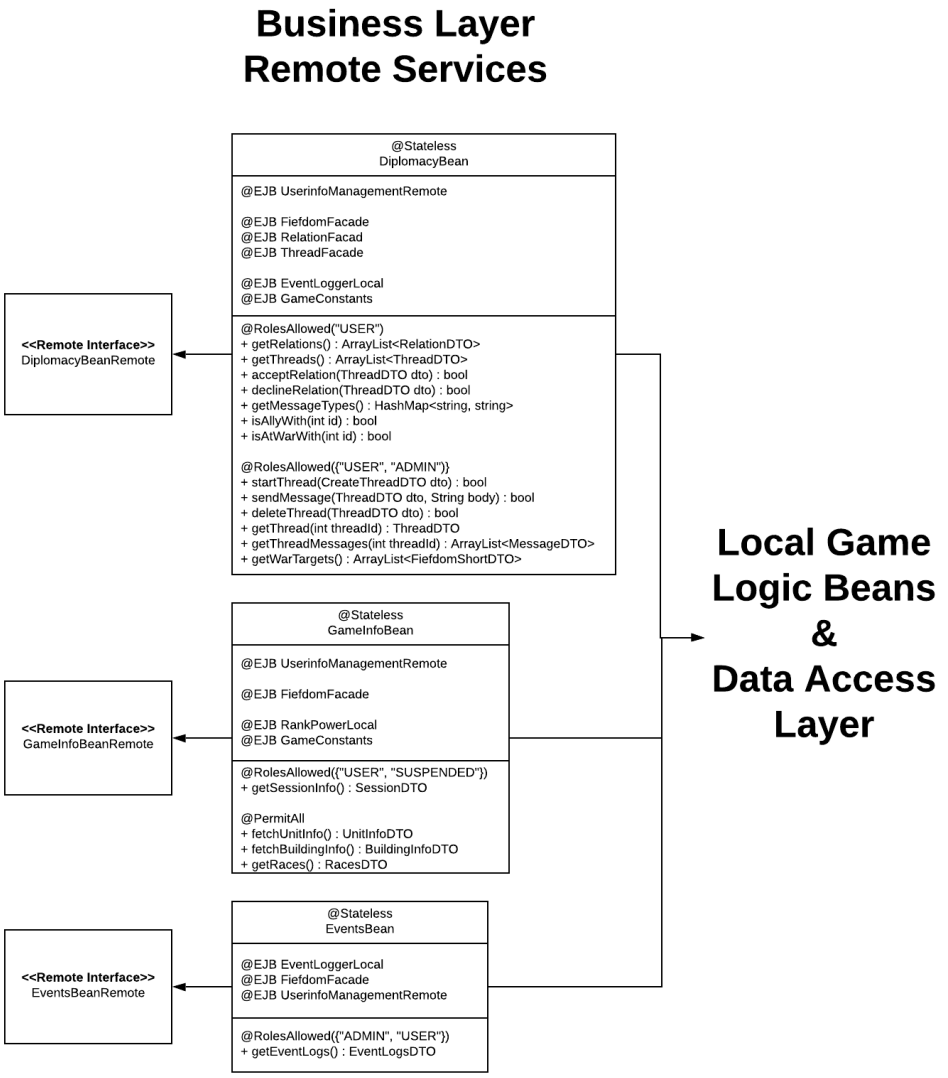


Figure 1.5 – Backing bean support for JSF views following SRP.

2. Business Layer

Two sub layers - front facing remote interfaces, and internal EJB's with local interfaces.

Stateless EJB's where possible to optimise scalability. Singletons used to cache non-critical data and avoid database hits. Concurrency via Java EE concurrency management.



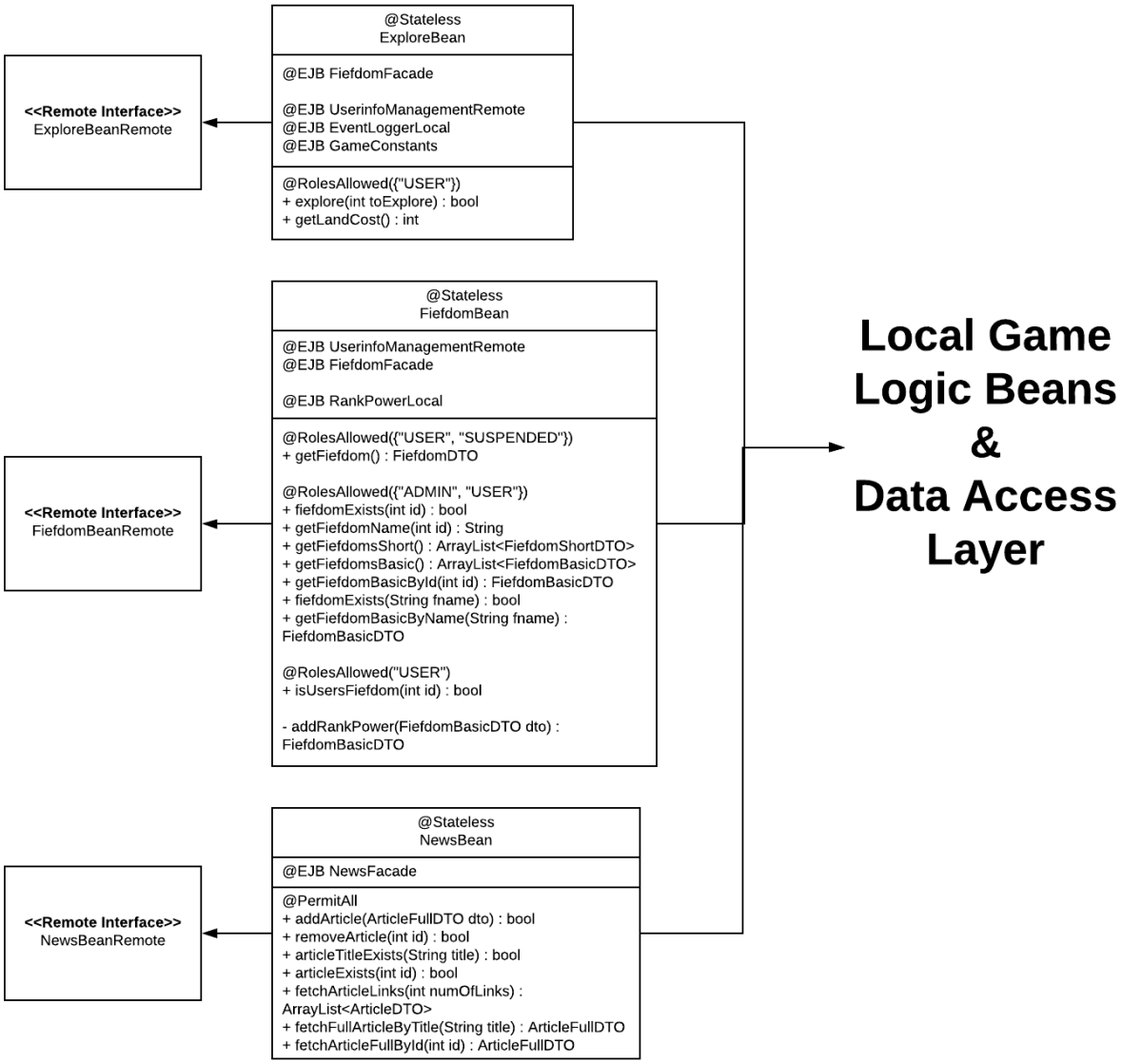


Figure 2.2a – Front facing business layer EJB's

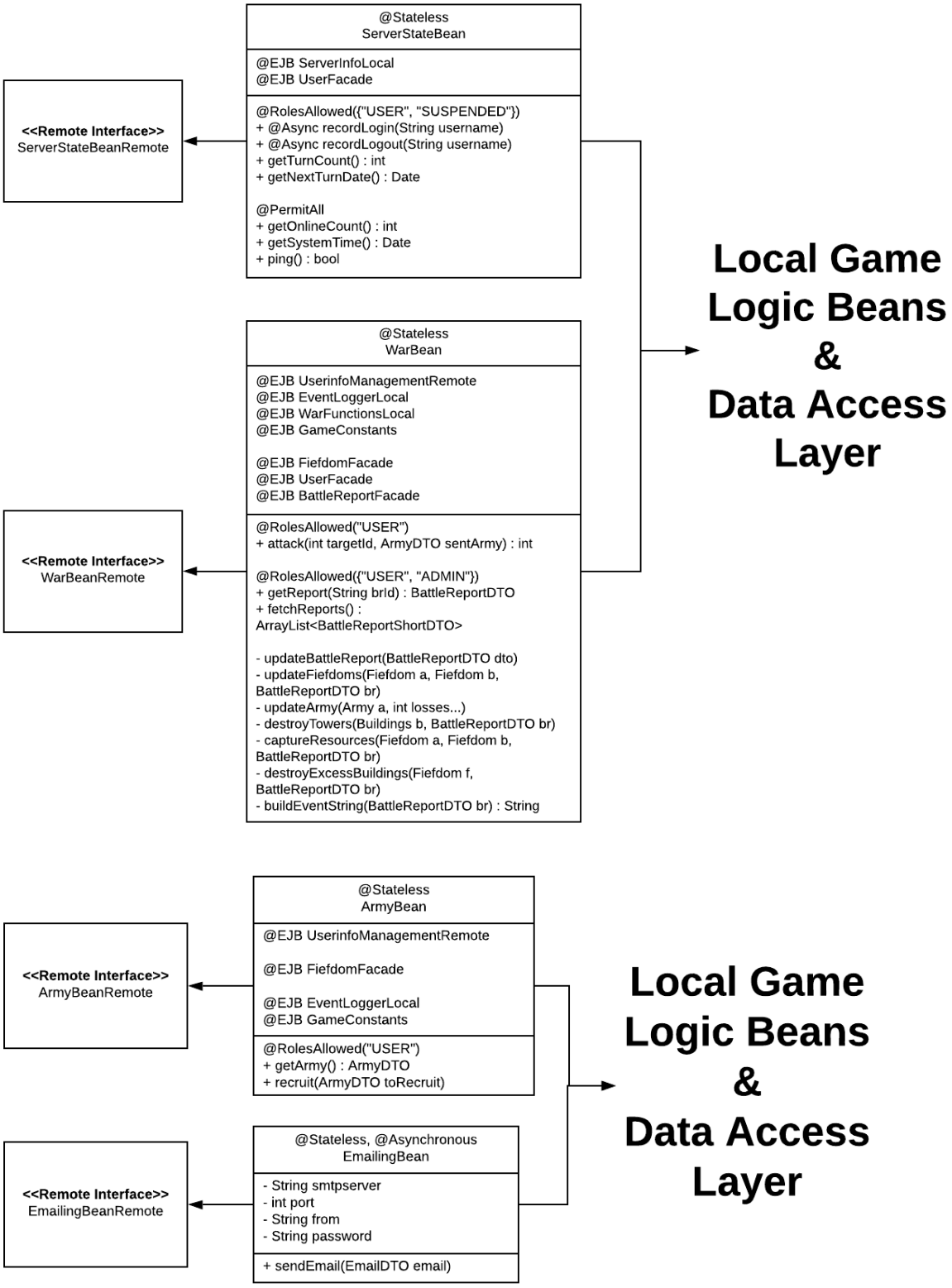


Figure 2.2b – Front facing business layer EJB's

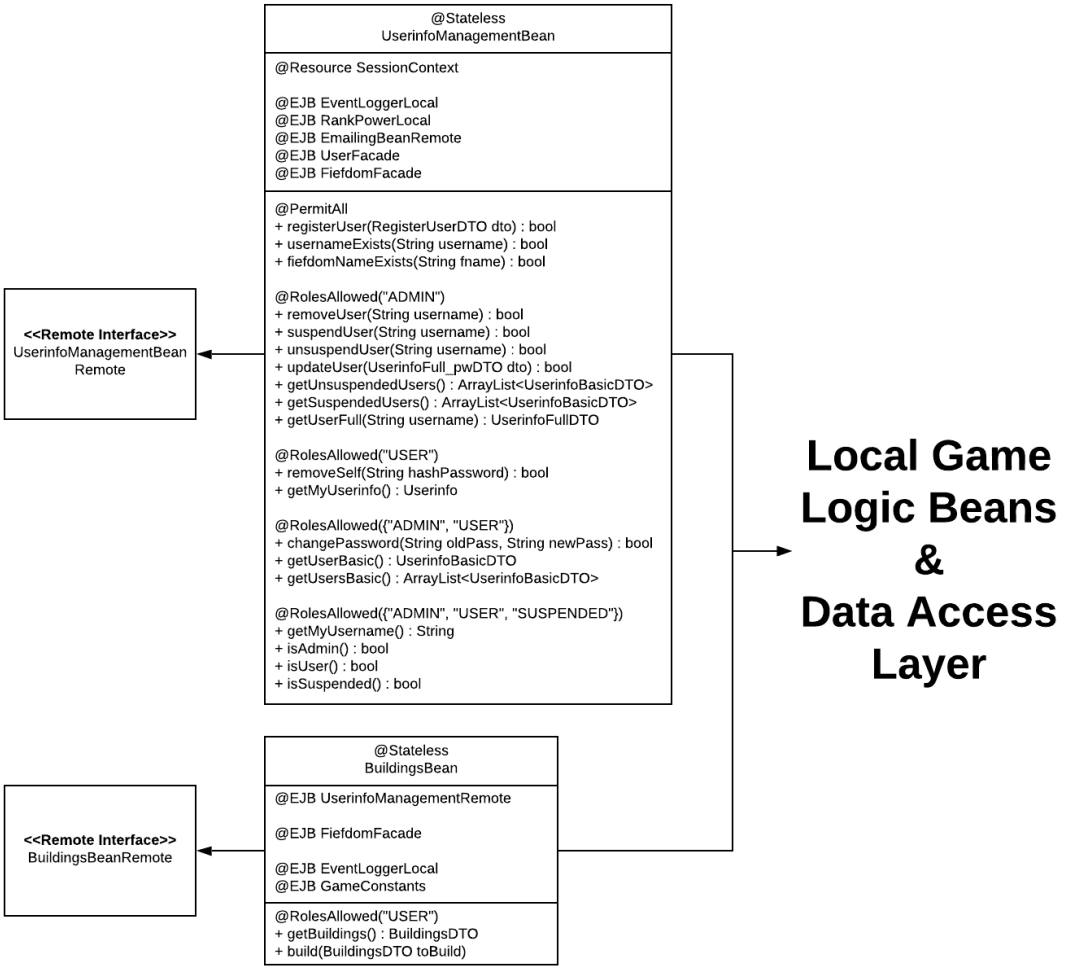
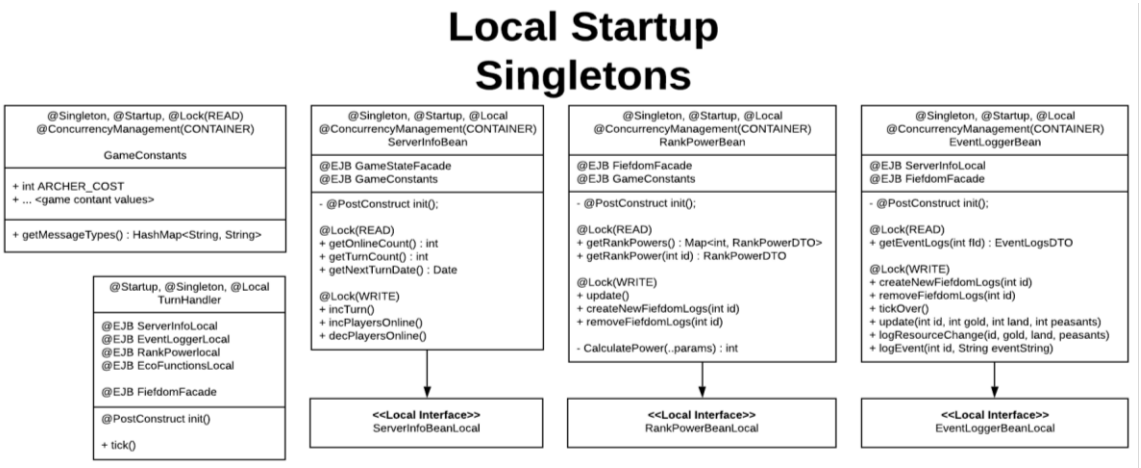
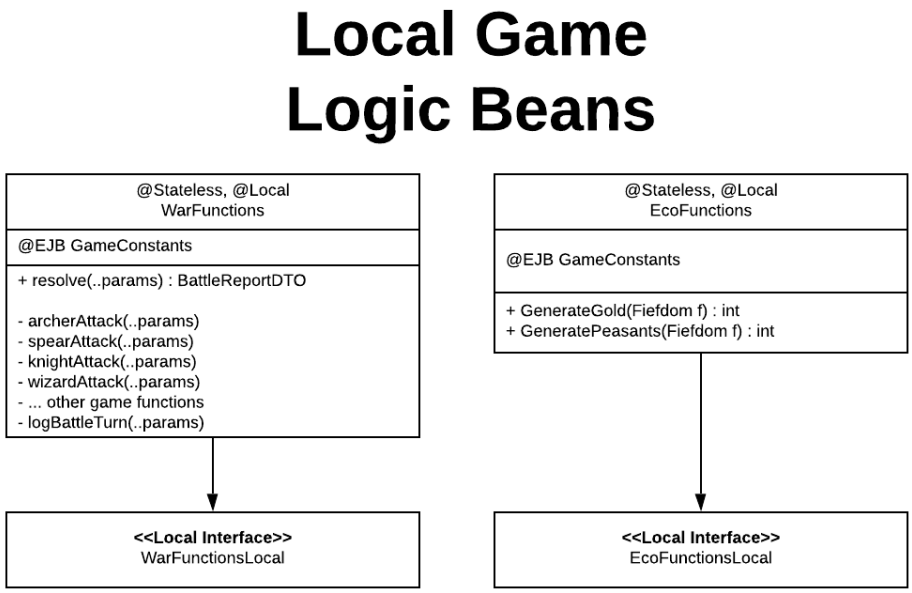


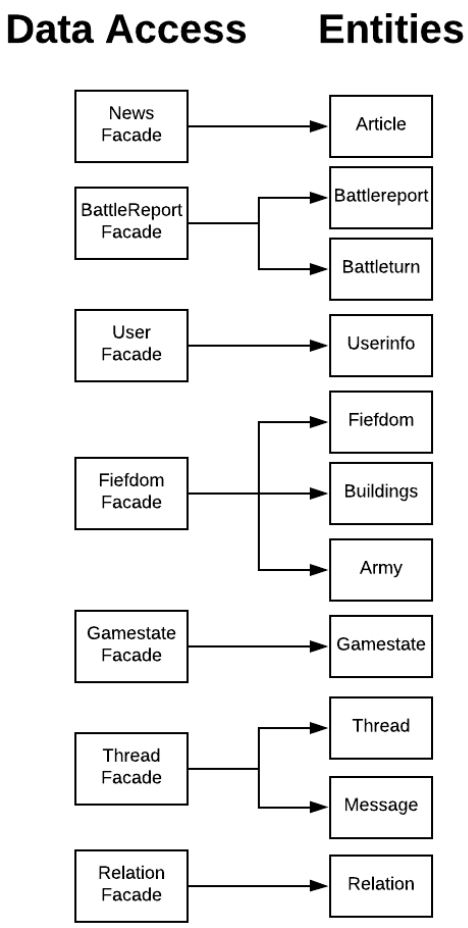
Figure 2.2c – Front facing Business Layer EJB's





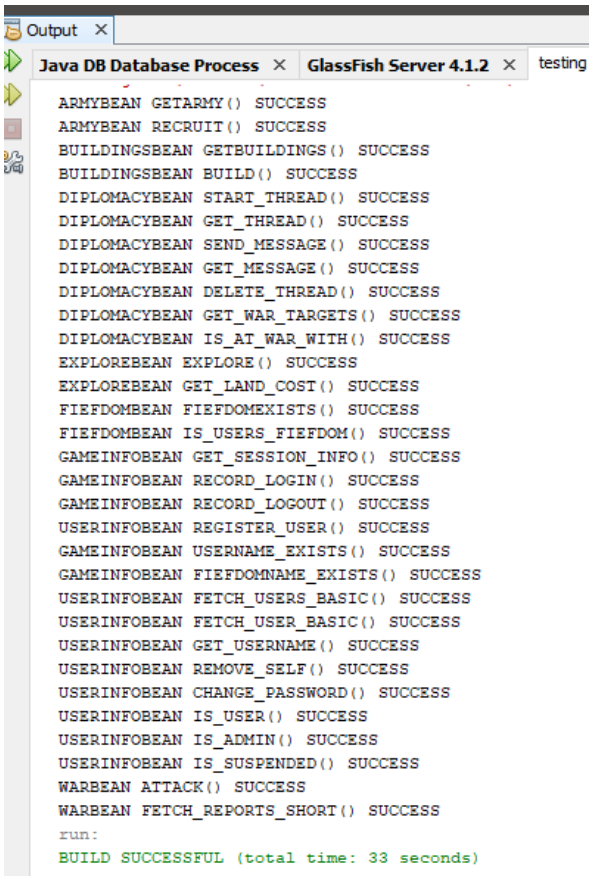
3. Data Access Layer

Object persistence via hibernate ORM / JPA via entity facades.



4. Software Testing Results

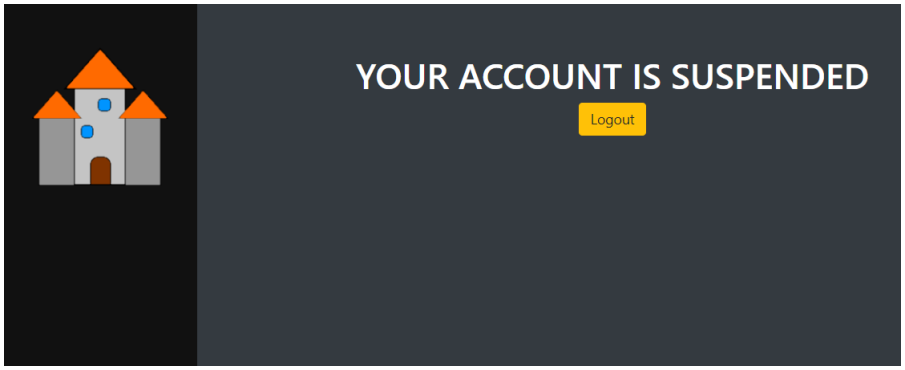
Testing harness over EJB remote interfaces.



Manual testing of JSF views and templating

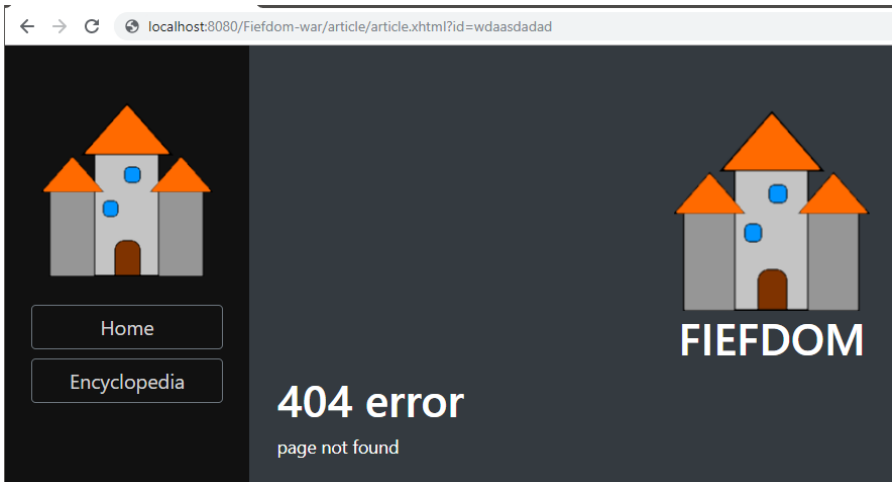
Testing Suspended user

1. Log in as Admin, suspend user, logout
2. Log in as suspended user

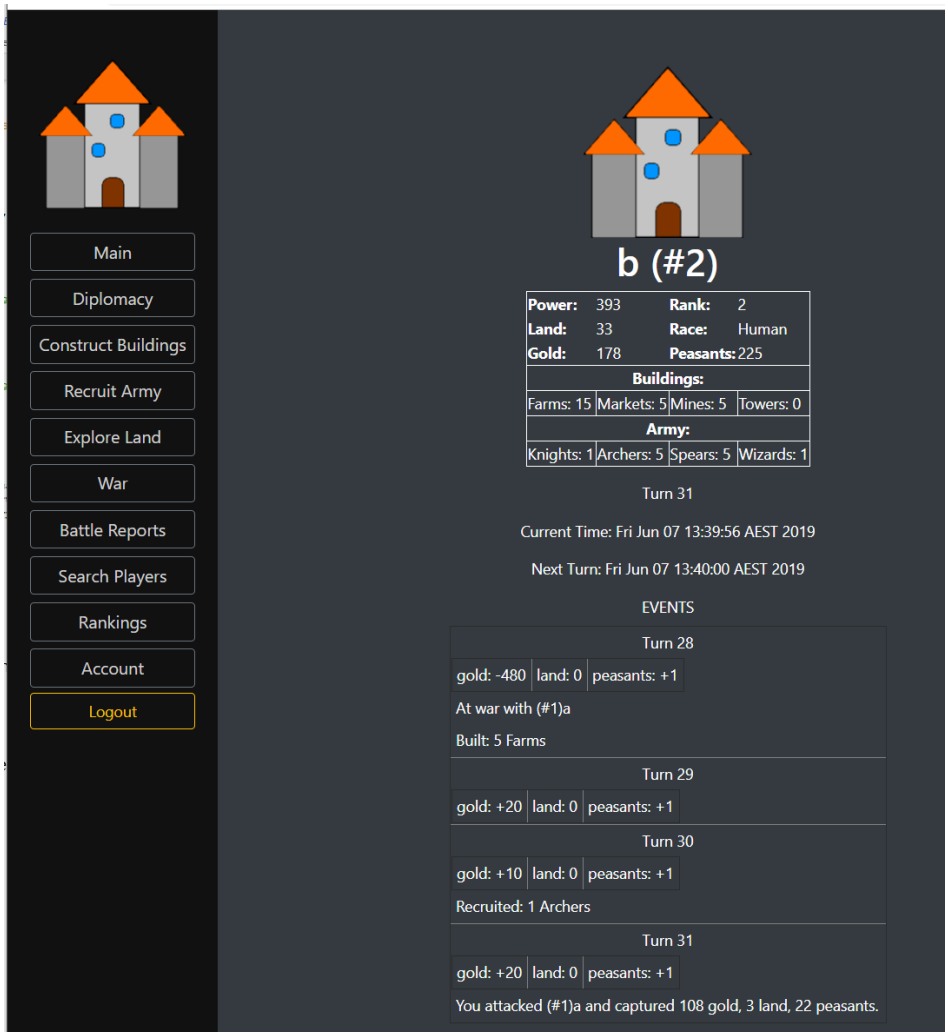


Test Invalid Requests

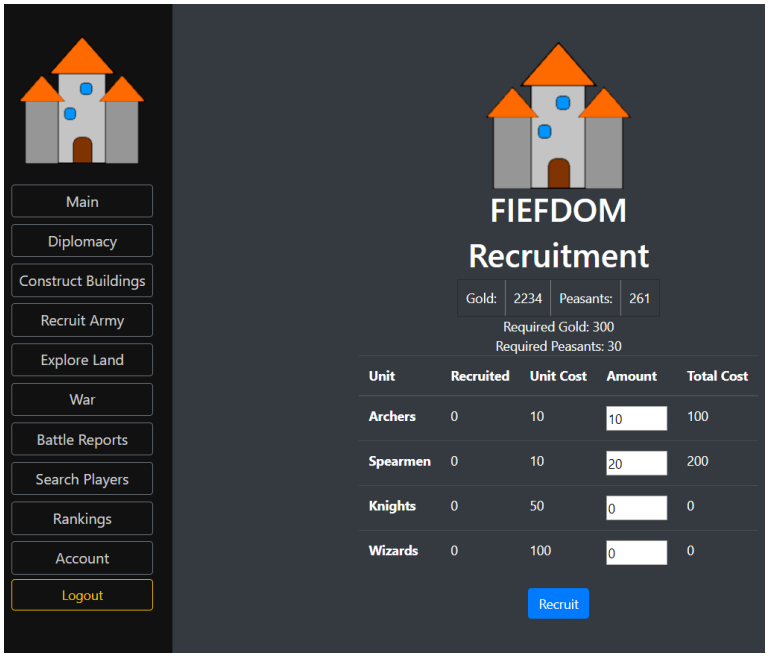
Invalid id for battle reports and article get requests



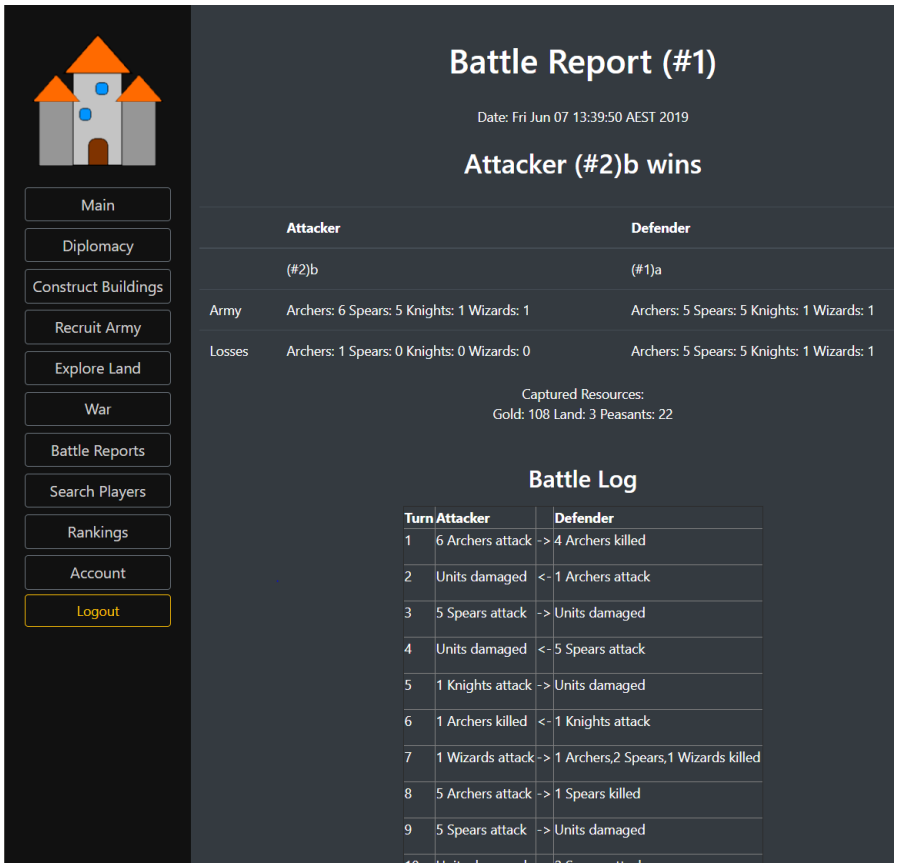
Test Event Logging

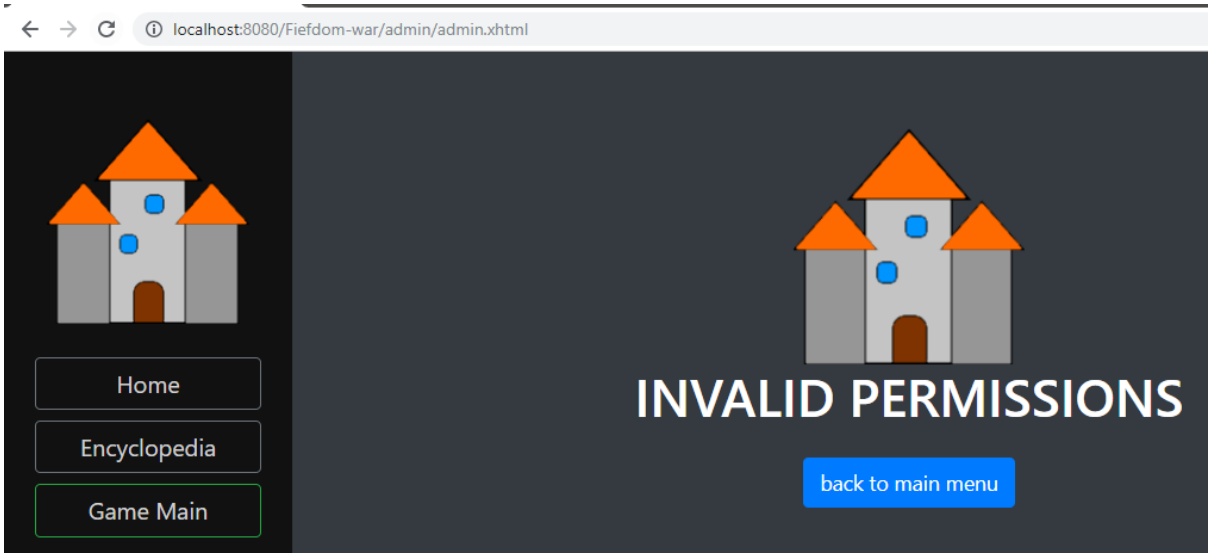


Test AJAX listeners



Test JSF formatting



Test Web Resource Security Constraints

Accessing restricted resources without appropriate authentication