**1. LOGIN**

screen->SBFClient

- sendLoginInfo()

SBFClient->screen

- displayLoginScreen()

- displayLoginSuccessScreen()

- displayLoginFailureScreen()

SBFClient->SBFServer

- validateUser(username:String, password:String)

SBFServer->SBFClient

- userValid()

- userInvalid()

**2. REGISTRATION**

screen->SBFClient

- requestSignUp()

- sendNewUserCredentials()

SBFClient->screen

- displayNewUserSignUpPage()

- displayNewUserSuccessPage()

- displayNewUserFailurePage()

SBFClient->SBFServer

- validateNewUserCredentials(username:string, password:string)

SBFServer->SBFClient

- newUserCredentialsValid()

- newUserCredentialsInvalid()

**3. REVIEW STATISTICS**

screen->SBFClient

- requestStatistics(player:String)

SBFClient->screen

- displayStatistics(stats:Statistics)

SBFClient -> SBFServer

- requestStatistics(player:String)

SBFServer -> SBFClient

- provideStatistics(stats:Statistics)

**4. GET MATCHED**

screen->SBFClient

- requestPlayers()

- challengePlayer(username:String)

SBFClient->screen

- displayAllPlayersPage(players:List)

- displayGameAcceptedPage()

- displayGameRejectedPage()

(the following is something of a clusterfuck and could definitely use revision)

SBFClient -> SBFServer

- requestAllCurrentPlayers()

- requestMatching(from:String, to: String)

- playerAcceptsGame(from:String, to:String)

- playerRejectsGame(from:String, to:String)

SBFServer -> SBFClient

- allCurrentPlayers(players:List)

- askIfPlayerWantsToJoin(to: String)

- gameRejected(to:String)

- gameAccepted(to:String)

**5. SET UP / GAME CONDITIONS / START GAME**

SBFClient->Screen

- displayGameConditions(conditions:GameConditions)

- acceptGameConditions()

- displayNewGame(conditions:Conditions)

screen->SBFClient

- sendNewConditions(conditions:GameConditions)

- sendConfirmationOfConditions()

SBFClient -> SBFServer

- submitGameConditions(conditions:GameConditions)

- submitAgreement()

SBFServer -> SBFClient

- newGameConditions(conditions:GameConditions)

- startGame(conditions:Conditions)

**6. LOAD / SAVE GAME**

screen->SBFClient

- saveGame(name:String)

- loadGame(name)

SBFClient->screen

- saveGameSuccessMessage()

- saveGameFailureMessage()

- listLoadableGames(names:List)

- loadGameSuccessMessage()

- loadGameFailureMessage()

SBFClient -> SBFServer

- saveGameState(name:String)

- loadGameState(name:String)

SBFServer -> SBFClient

- saveGameSuccess()

- saveGameFailure()

- loadGameSuccess()

- loadGameFailure()

**7. SELECT SHIPS**

screen->SBFClient

- select(ship:Ship)

- deselect(ship:Ship)

- submitShipSelections()

SBFClient->screen

- displayShipSelectionScreen(ships:List)

SBFClient -> SBFServer

- validateShipSelection(ships:List)

SBFServer -> SBFClient

- listSelectableShips(ships:List)

- shipSelectionValid()

- shipSelectionInvalid()

**8. PLACE SHIPS**

screen->SBFClient

- placeShip(ship:Ship)

- removeShip(ship:Ship)

- submitShipPlacement()

SBFClient ->screen

- displayPlaceShipsPage()

- displayInvalidPlacementPage()

- displayValidPlacementPage()

SBFClient -> SBFServer

- alertOfShipPlacement(shipPlacements:List)

SBFServer -> SBFClient

- placementInvalid()

- placementValid()

**9. MAKE ACTION (other than moving ships)**

screen->SBFClient

- makeAction(action:InGameAction)

SBFClient ->screen

- displayActionInvalidPage()

SBFClient -> SBFServer

- submitAction(action:InGameAction)

SBFServer -> SBFClient

- actionValid()

- actionInvalid()

**10. UPDATING GAME SCREEN BETWEEN PLAYS**

screen -> SBFClient

-

SBFClient ->screen

- displayNewGameScreen()

SBFClient -> SBFServer

-

SBFServer -> SBFClient

- newGameScreen(shipPlacement:List, shipDamage:List)