// ---------------- Detailed Design Start – Workflows --------------------------

Start Application Workflow –-------------------------------------------

This is the code that gets run when the app starts, before any admin input.

* Initialize Firebase, picking appropriate db environment
* Get Game Data for the single, pre-set game. Set global variables including game status.
* Subscribe to Game Data changes

Game Data document listener onSnapShot Workflow (Subscribe)

* Update screen data with game status
* Game Status check
  + Paused
    - If I’m active, move myself to waiting queue
    - Update my status in the db
    - Set waiting queue to only this player. Otherwise, wouldn’t be paused status if any other players exist.
  + Game Not Started – unlikely status, resetting game?
    - Log off user
  + Completed – update screen. Still allow user to log in and out and view bounties, but nothing else.

Add Player Workflow --------------------------------------------------------------

* If game status is completed, reject add player request
* Get screen data
* Get player ref. If player already exists, reject add player request
* Else continue
  + Add player to db
  + Add player to waiting queue

Approve Player Workflow – This is either approving a registered player or a player that buys back in, in person with the admin. -----------------------------------------------

* Check if Game status is completed. If yes, reject, if not, continue
* Get screen data – id
* Get player ref. If not exist, reject, else continue
* Check player status – If inactive or registered, continue, else reject
* Set player status in db to waiting
* Put player in waiting queue – 2 instances, may be worth creating a function.

Approve All Players Workflow – This allows the admin to approve all registered players with one click. -----------------------------------------------

* Check if Game status is completed. If yes, reject, if not, continue
* Query db for all registered players, store in snapshot.
* Get the Waiting queue. Save into tempQueue.
* For each player in snapshot:
  + Add player to tempQueue
  + Update player status to Waiting in the db
* Write tempQueue to Waiting queue in db

Start Game Workflow –-------------------------------------------

This is the code that moves all Waiting players to Active. Can be used to start the game or resume the game after a Paused scenario.

* Check game status, only allow if status = not started or paused.
* Get the waiting queue. Loop:
  + Create Chain link
  + Activate player
* Delete waiting queue
* Update screen

Pay Bounties Workflow -----------------------------------------------

* Get and check screen data – id, bounties data
* Get player ref. If not exist, reject, else continue
* Check bounty data on screen vs. db, if valid continue, else reject and report.
* Update players bounties owed in the db

Check Bounties Workflow -----------------------------------------------

* Get screen data – id
* Get player ref. If not exist, reject, else continue
* Retrieve bounty data from db and display on screen

Bomb Button Workflow ----------------------------------------------------

* Check game status, if Active, then continue, else reject
* Get chain and store in tempChain
* Get waiting queue and add to tempChain, also store waiting player ids in tempQueue
* Shuffle all players in tempChain
* Create new chain
  + Check if player added to tempChain was in tempQueue, need to then flip their status to Active in the db. Else, player was already Active, do nothing.

Upload Player Picture Workflow --------------------------------------

* Get screen data – id, if id not blank, continue, else reject
* Retrieve file data from the screen
* If the number of files selected is 1 then proceed, else reject
  + Get a reference to the Player document
  + Update player picture name field
  + Create the full path variable using the players id.
  + Upload the file to cloud storage
  + Monitor the upload and log the status

View Player Picture Workflow -----------------------------------------

* Get screen data – id
* Get a reference to the Player object for the picture name field
* Create the full path variable for the file download using the players id and picture name
* Attempt to download picture file. If success:
  + Update picture on screen
* Else failure or missing picture – Display correct error.

Search Player Workflow -----------------------------------------

* Get screen data – id and name
* Query db for name first if entered
  + If found – Loop through each player found with that name and display their name and id on the screen
  + If not found – show not found
* If Name not entered, but id entered
  + Get player ref based on id
  + If doc exists, player exists, show data on screen. No need to loop, only 1 will exist by id.

Remove Player Workflow -----------------------------------------

* Get screen data – id
* Get player ref based on id
* Switch on player status if found:
  + Active
    - **Update Chain to bypass me** (See Player Design)
  + Waiting
    - Remove player from Waiting queue on db
  + Any other status
    - Do nothing
* Delete player document

Volunteer Button Workflow ---------------------------------

* Set volunteer needed data field in the db to true. Subscribers do the rest.

End Game Workflow -----------------------------------------

* Set game status to Completed in db
* Delete queue
* Delete chain
* Rely on Game Status listeners on Player app.