

## Capstone Project Phase 1:

**Team:** Marco Lauguico, Peter Tang, Kevin Do

**Project Name:** Smite Item Builder

**Purpose:** The game Smite has a lot of characters with different stats, and with that they are able to build 6 items. The game has hundreds of items to choose through, and this application will allow the user to test out different build combinations to best suit what they want. The app will have everything from god stats, item stats, item descriptions, lore and much more.

**Audience:** Users who play the game Smite whether it be new players or professionals

**Non-Profit:** May release it, need intent of teammates.

### Database Listing:

Gods																	
Id	Name	Class	Type	Art	Base Health	Base Physical Protection	Base Magical Protection	Base Mana	Base Attack Speed	Base HP5	Base MP5	Physical Protection Scaling	Magical Protection Scaling	Base Movement Speed	Base Damage	Damage Scaling	Pantheon
Integer Primary Key	TEXT	TEXT	TEXT	BLOB	INTEGER	INTEGER	INTEGER	INTEGER	REAL	REAL	REAL	REAL	REAL	INTEGER	INTEGER	REAL	TEXT

Items																			
Id	Item Name	Art	Cost	Type	Health	Mana	Physical Protection	Magical Protection	Power	Attack Speed	Percent Penetration	Flat Penetration	Critical Chance	Cooldown Reduction	Lifesteal	Passive	HP5	MP5	Crowd Control Reduction
Integer Primary Key																			
	TEXT	BLOB	INTEGER	TEXT	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	REAL	REAL	INTEGER	REAL	INTEGER	REAL	TEXT	REAL	REAL	INTEGER

NPC											
Id	NPC Name	Art	Health	Physical Protection	Magical Protection	Power	Type	Damage vs Towers	EXP Reward	Gold Reward	
Integer Primary Key	TEXT	BLOB	INTEGER	INTEGER	INTEGER	INTEGER	TEXT	INTEGER	INTEGER	INTEGER	
Builds											
Id	God	Item1	Item2	Item3	Item4	Item5	Item6	Relic1	Relic2		
Integer Primary Key	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER		
	Relates to God table	Relates to Items table	Relates to Items table	Relates to Items table	Relates to Items table	Relates to Items table	Relates to Items table	Relates to Relic Table	Relates to Relic Table		

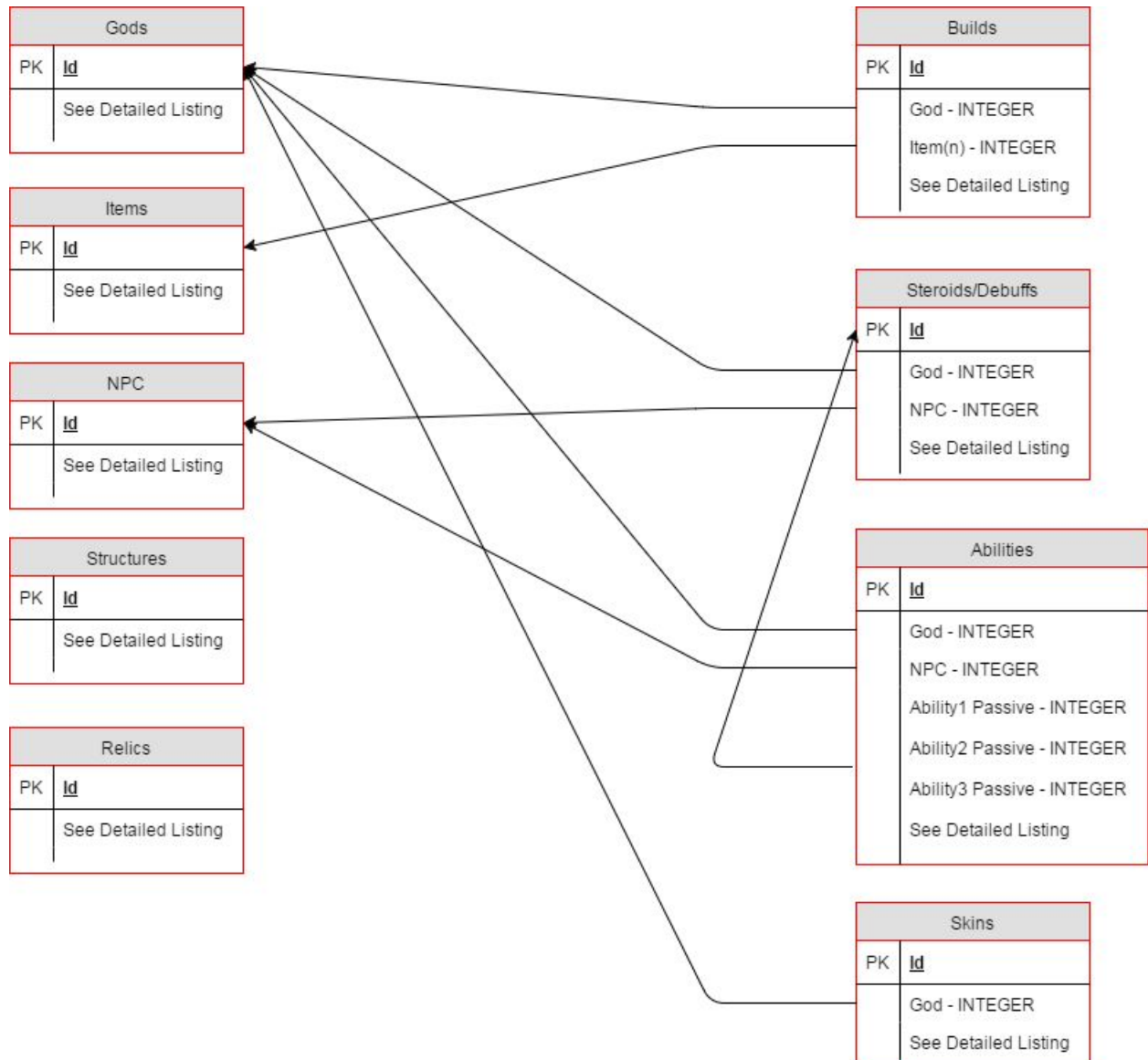
Structures											
Id	Name	Art	Class	Health	Physical Protections	Magical Protections	Base Damage	Gold Worth	Respawnable		
Integer Primary Key	TEXT	BLOB	TEXT	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER		
Steroids/Debuffs											
Id	God	Art	NPC	Duration	Power	Movement Speed	Lifesteal	Protections	HP5	MP5	Penetration
Integer Primary Key	INTEGER	BLOB	INTEGER	REAL	INTEGER	REAL	INTEGER	INTEGER	REAL	REAL	INTEGER
	Relates to Gods table		Relates to NPC Table								

Abilities																			
Id	God	Ability1 Name	Ability1 Damage	Ability1 Scaling	Ability1 Overtime	Ability1 Passive	Ability2 Name	Ability2 Damage	Ability2 Scaling	Ability2 Overtime	Ability2 Passive	Ability3 Name	Ability3 Damage	Ability3 Scaling	Ability3 Overtime	Ability3 Passive	Ultimate Name	Ultimate Damage	Ultimate Scaling
Integer Primary Key	INTEGER	TEXT	INTEGER	REAL	INTEGER	INTEGER	TEXT	INTEGER	REAL	INTEGER	INTEGER	TEXT	INTEGER	REAL	INTEGER	INTEGER	TEXT	INTEGER	REAL
	Relates to Gods table					Relates to Steroids/ Debuffs table					Relates to Steroids/ Debuffs table					Relates to Steroids/ Debuffs table			

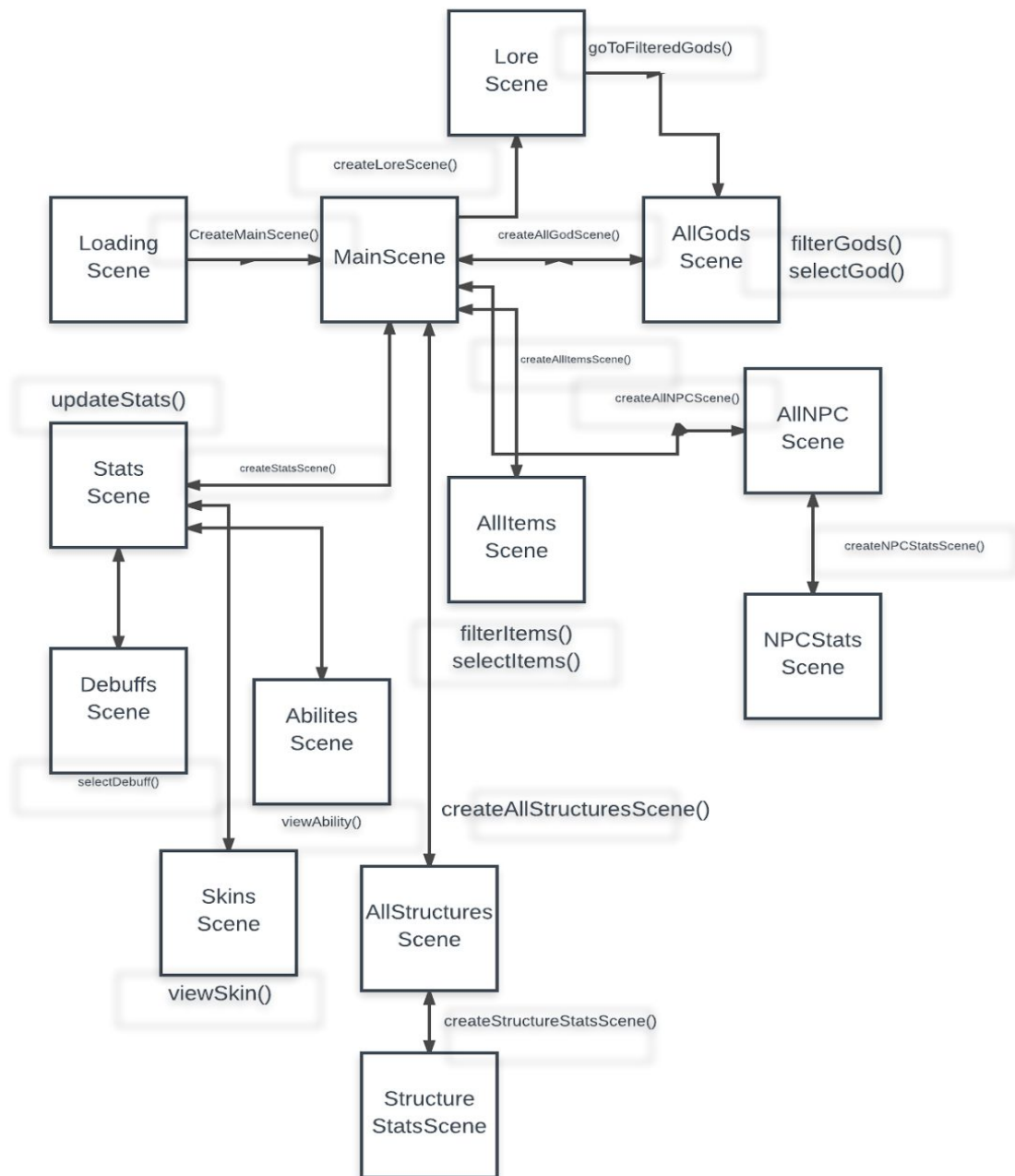
Skins				
Id	God	Tier	GemCost	CardArt
Integer Primary Key	INTEGER	INTEGER	INTEGER	BLOB
	Relates to Gods table			

Relics				
Id	Relic Name	Art	Ability	Upgraded
Integer Primary Key	TEXT	BLOB	TEXT	INTEGER

## Database Schema:

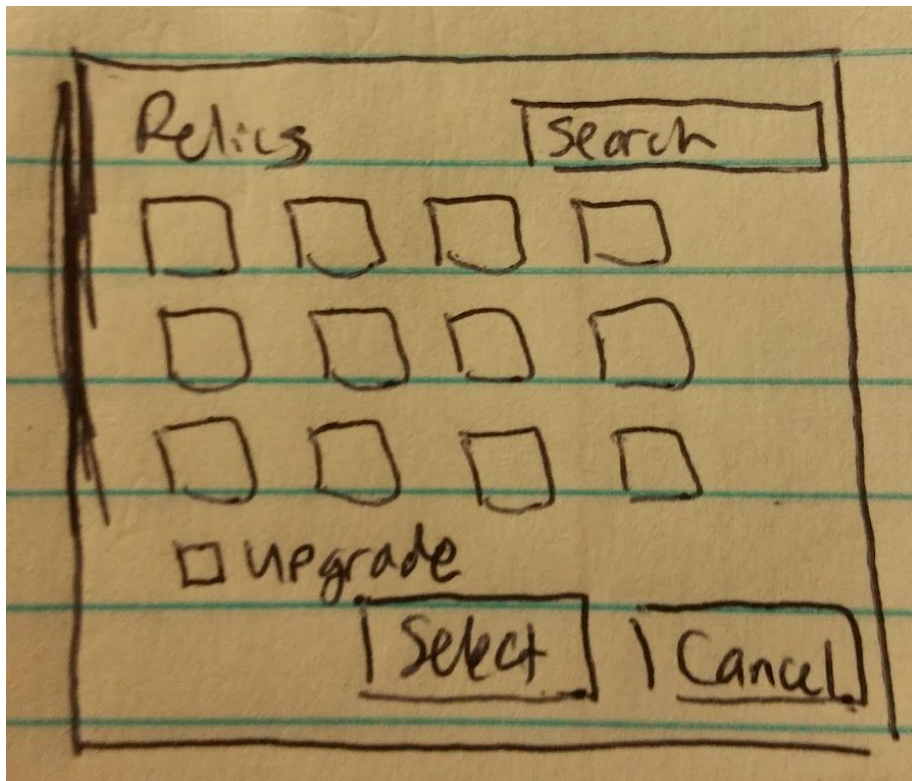


## Scene Flow

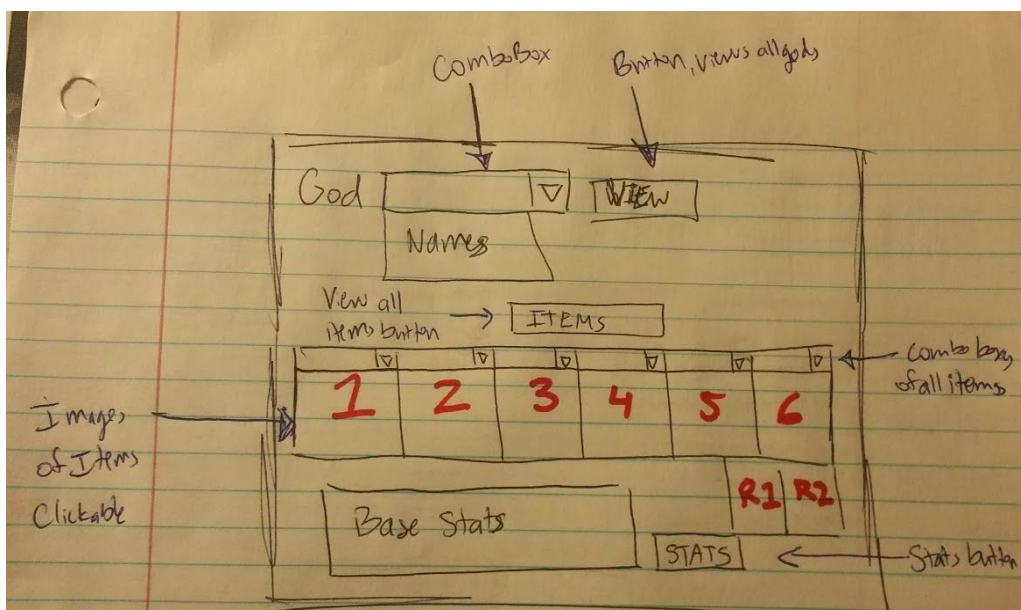


## Examples of Scenes:

Relics Scene -> shows all relics, can choose to add to build

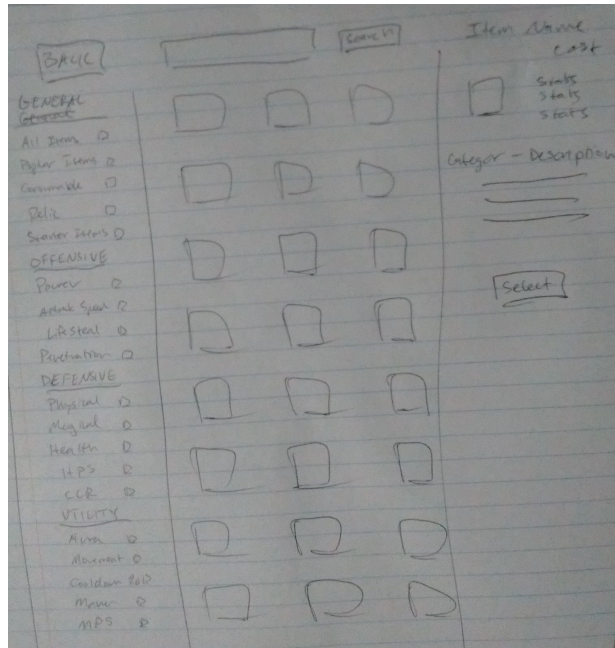


Main Screen -> View all gods, view god stats, view all items, view lore





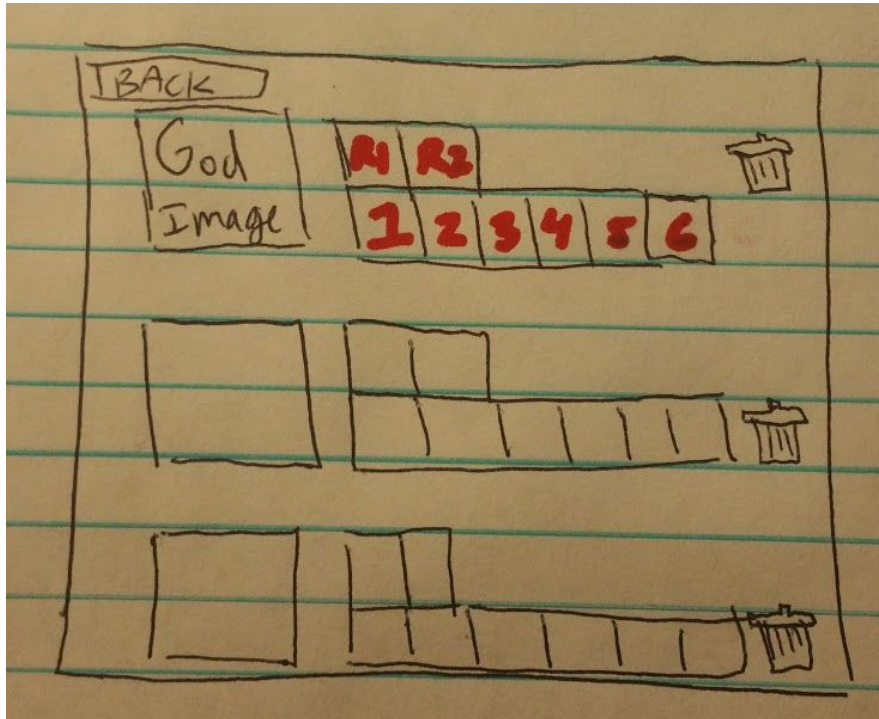
All Items Page -> Filter items, Checkboxes for different types of items, Back to main screen



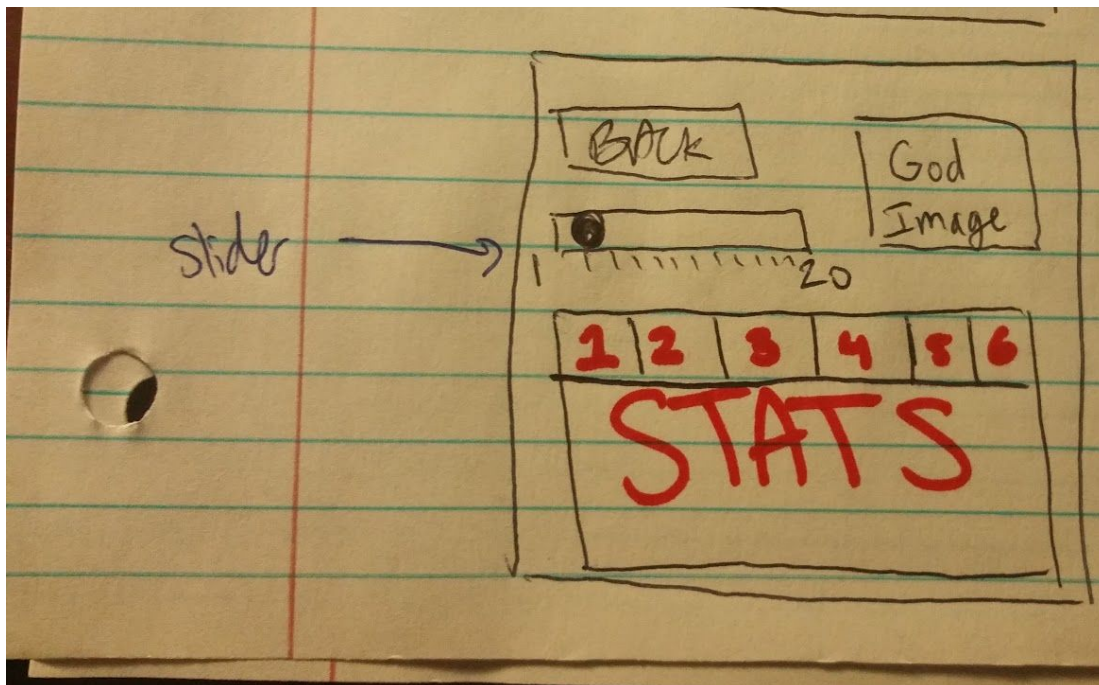
Essentially a close adaptation of -->



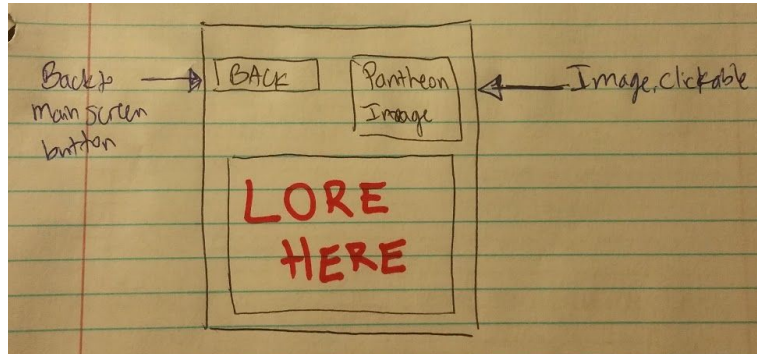
View Saved Builds -> load saved builds, delete builds, search builds, Back to main screen



Stats page -> Update stats based on Lv (from slider), Back to main screen



Lore page -> clickable pantheon logo shows all gods of that Pantheon, Back to main screen



## Activities

- `viewAllGods()` - this is an activity that will bring the user to a list of all god icons on which they can press the god's icon and once they did, it'll bring them to a scene where the God's attributes are displayed. This list comes from the model (database), creates tables from database if not already done so
- `viewAllItems()` - this is an activity that will prompt the user to a list of all item icons on which they can press and once they did, it'll prompt them to a scene where the item's description and item effects are displayed. This list comes from the model (database), creates tables from database if not already done so
- `viewGod()` - this is a scene where it displays a God's attributes. \*Name, Description(lore), Stats, and Abilities. Pulls the info from the model (database), creates tables from database if not already done so
- `viewItem()` - this is a scene where it displays an item's description and effects. Pulls data from the model (database), creates tables from database if not already done so
- `filterItems()` - filters out list of all items per checkbox parameters
- `searchItem()` - searches subsequence in item name with all filters applied
- `viewAllBuilds()` - this is an activity inside the `viewGod()` scene where if is active will bring the user to the scene where all the list of all possible builds for that certain God is displayed.
- `addBuildScene()` - this is an activity where if a user activate it will bring the user to a scene where they can name the build and etc and add it on the build database.
- `addBuild()` - this is an activity that creates a new record of a build for a God in the build database.



- deleteBuild() - this is an activity that removes a God build record from the build database.
- selectItem() - this selects the items, takes the user to the main screen, adds it to the build and updates all stats accordingly
- selectGod() - this selects the god, takes the user to the main screen and updates all stats
- selectRelic() - this selects the relic, takes the user to the main screen, and adds it to the build
- editBuild() - this is an activity that will prompt the user to the addBuildScene() and will load the previous values for the build and the user can edit them and updates it on the build database.
- viewBuildScene() - this is an activity where if activated will bring the user to a scene where the stats and the item build of a God are shown and a slider from (1-25) that will show the stats on a certain level.
- sliderStatHandler() - this activity is connected to the viewBuildScene() where there is a slider from 1-25 and will update the displayed stats from a certain God according to the level.

Model	Controller	View
<p>All Database tables above</p> <ul style="list-style-type: none"> <li>• All entity class member variables mirror its corresponding database attributes</li> </ul> <p><b>Gods.java</b></p> <ul style="list-style-type: none"> <li>• mId</li> <li>• mName</li> <li>• mClass</li> <li>• mType</li> <li>• mArt</li> <li>• mStats</li> <li>• mBuilds (An arraylist/dynamic array of build class)</li> <li>• mPantheon</li> </ul> <p><b>Items.java</b></p> <ul style="list-style-type: none"> <li>• mId</li> <li>• mName</li> <li>• mArt</li> </ul>	<ul style="list-style-type: none"> <li>• Singleton Pattern</li> <li>• So far planning to go with one controller</li> <li>• Examples of methods within the controller to process and return result to the View</li> </ul> <ul style="list-style-type: none"> <li>• viewAllGods()</li> <li>• viewGod()</li> <li>• searchGod()</li> <li>• filterGods()</li> </ul> <ul style="list-style-type: none"> <li>• viewAllItems()</li> <li>• filterItems()</li> <li>• viewItem()</li> <li>• searchItem()</li> <li>• selectItem()</li> </ul>	<ul style="list-style-type: none"> <li>• JavaFX application</li> <li>• Scenes listed by scene name</li> </ul> <p>Splash/Title/Loading Scene</p> <p>AllGodsScene</p> <ul style="list-style-type: none"> <li>• View with all gods selection</li> </ul> <p>AllItemsScene</p> <ul style="list-style-type: none"> <li>• View with icons of all items</li> </ul>

<ul style="list-style-type: none"> <li>• mCost</li> <li>• mType</li> <li>• mStats</li> </ul> <b>Relics.java</b> <ul style="list-style-type: none"> <li>• mId</li> <li>• mName</li> <li>• mArt</li> <li>• mAbility</li> <li>• mIsUpgraded</li> </ul> <b>Structures.java</b> <ul style="list-style-type: none"> <li>• mId</li> <li>• mName</li> <li>• mArt</li> <li>• mClass</li> <li>• mHealth</li> <li>• mPhysicalProt</li> <li>• mMagicalProt</li> <li>• mBaseDamage</li> <li>• mGoldWorth</li> <li>• mIsRespawnable</li> </ul> <b>Npc.java</b> <ul style="list-style-type: none"> <li>• mId</li> <li>• mName</li> <li>• mArt</li> <li>• mStats</li> <li>• mExp Gain</li> <li>• mGold Gain</li> </ul> <b>SteroidsDebuffs.java</b> <ul style="list-style-type: none"> <li>• mId</li> <li>• mGod</li> <li>• mArt</li> <li>• mNPC</li> <li>• mDuration</li> <li>• mPower</li> <li>• mMoveSpeed</li> <li>• mLifesteal</li> <li>• mProtection</li> <li>• mHP5</li> <li>• mMP5</li> <li>• mPenetration</li> </ul> <b>Builds.java</b> <ul style="list-style-type: none"> <li>• Id</li> <li>• God</li> <li>• inventory</li> </ul> <b>Abilities.java</b> <ul style="list-style-type: none"> <li>• See above DB description</li> </ul>	<ul style="list-style-type: none"> <li>• viewBuild()</li> <li>• sliderStatHandler()</li> <li>• searchBuild()</li> </ul> <ul style="list-style-type: none"> <li>• viewAllRelics()</li> <li>• selectRelic()</li> <li>• filterRelics()</li> <li>• upgradeRelic()</li> </ul> <ul style="list-style-type: none"> <li>• addBuild()</li> <li>• editBuild()</li> <li>• deleteBuild()</li> </ul> <ul style="list-style-type: none"> <li>• viewLore()</li> </ul> <ul style="list-style-type: none"> <li>• Remaining scenes follow a template to view all items of the game's attribute with the ability to filter through the list and show the filtered results</li> </ul>	<b>StatsScene</b> <ul style="list-style-type: none"> <li>• View of God's stats</li> </ul> <b>RelicsScene</b> <ul style="list-style-type: none"> <li>• View with all relics</li> </ul> <b>SavedBuildsScene</b> <ul style="list-style-type: none"> <li>• View with editable build</li> </ul> <b>LoreScene</b> <ul style="list-style-type: none"> <li>• View with God's lore</li> </ul> <b>AllNPCScene</b> <b>NPCStatsScene</b> <b>DebuffScene</b> <b>AbilitiesScene</b> <b>AllStructuresScene</b> <b>StructureStatsScene</b> <b>SkinsScene</b> etc.
---	---	--