Evidence for Analysis and Design Unit
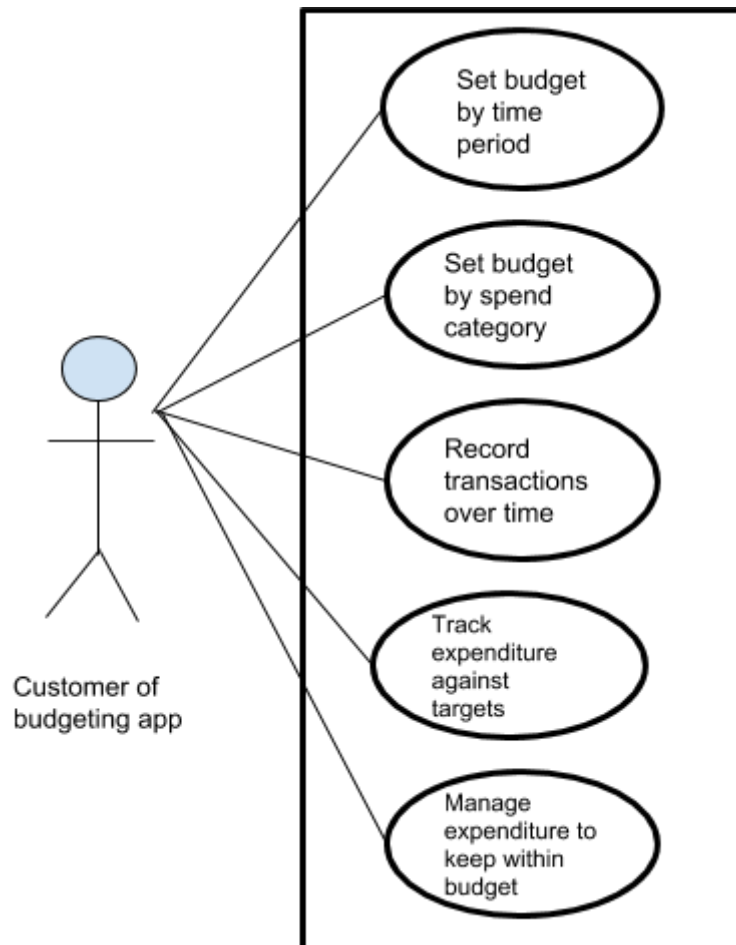
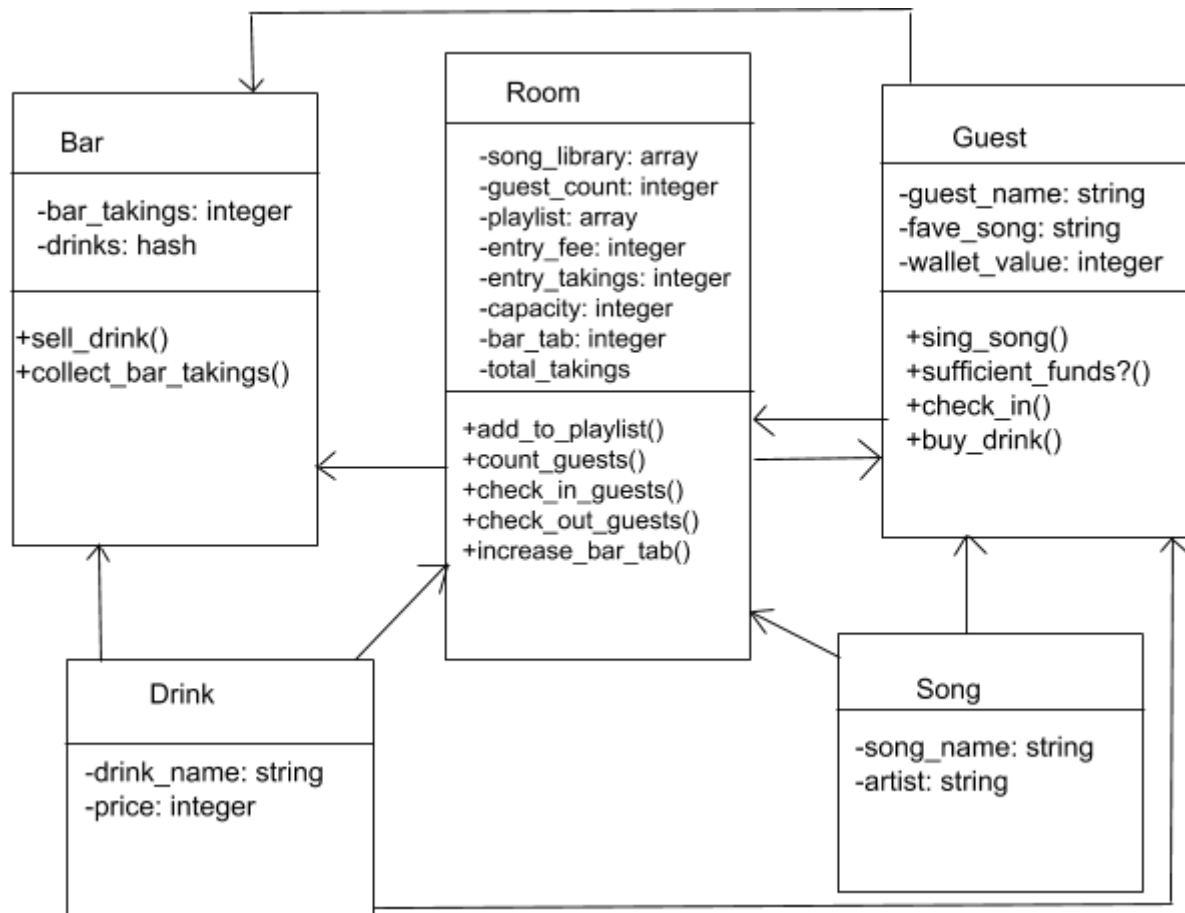**Verity Ashforth**
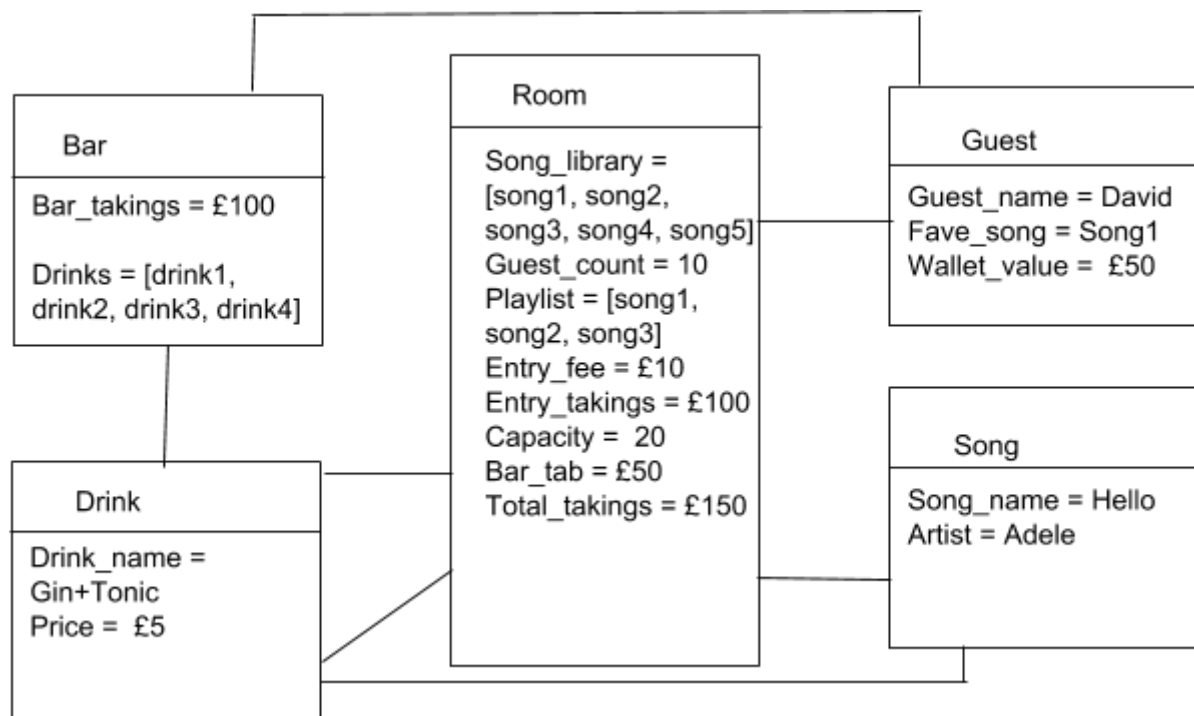**E19**

AD1 - Use Case Diagram
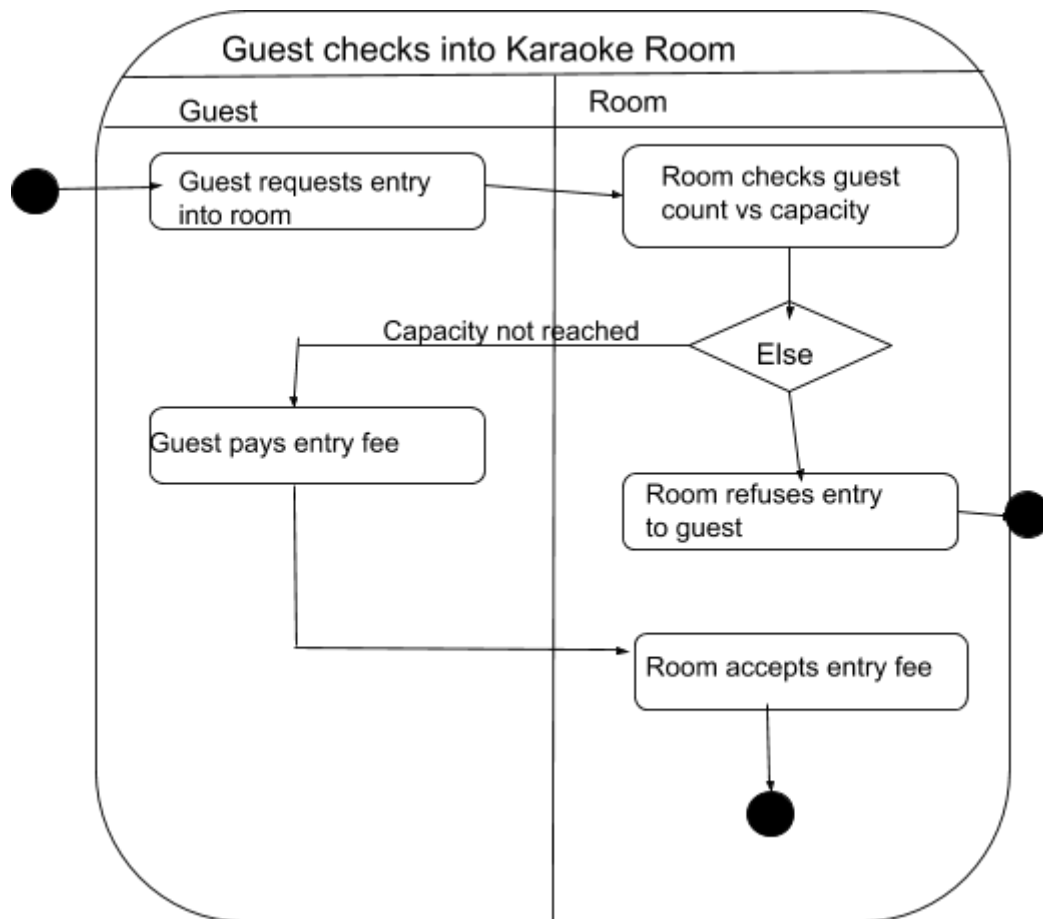

AD2 - A Class Diagram

Bar

-bar_takings: integer
-drinks: hash

+sell_drink()
+collect_bar_takings()

Room

-song_library: array
-guest_count: integer
-playlist: array
-entry_fee: integer
-entry_takings: integer
-capacity: integer
-bar_tab: integer
-total_takings

+add_to_playlist()
+count_guests()
+check_in_guests()
+check_out_guests()
+increase_bar_tab()

Guest

-guest_name: string
-fave_song: string
-wallet_value: integer

+sing_song()
+sufficient_funds?()
+check_in()
+buy_drink()

Drink

-drink_name: string
-price: integer

Song

-song_name: string
-artist: string

AD3 - An Object Diagram

Bar

Bar_takings = £100

Drinks = [drink1, drink2, drink3, drink4]

Room

Song_library = [song1, song2, song3, song4, song5]
Guest_count = 10
Playlist = [song1, song2, song3]
Entry_fee = £10
Entry_takings = £100
Capacity = 20
Bar_tab = £50
Total_takings = £150

Guest

Guest_name = David
Fave_song = Song1
Wallet_value = £50

Song

Song_name = Hello
Artist = Adele

Drink

Drink_name = Gin+Tonic
Price = £5

AD4 - Activity Diagram

AD5 - An Inheritance Diagram

Animal

- number of eyes : int
- name: String
- age: int

- canEat()
- canSleep()
- canReproduce()

Dog

Inherited properties
- number of eyes : int
- name: String
- age: int

Own properties
- teeth : int
- tail size : int

Inherited methods
- canEat()
- canSleep()
- canReproduce()

Own methods
- can bark()

Flamingo

Inherited properties
- number of eyes : int
- name: String
- age: int

Own properties
- feather colour: String
- beak size: int

Inherited methods
- canEat()
- canSleep()
- canReproduce()

Own methods
- canBalanceOnOneLeg()

AD6 - Implementation Constraints

Topic	Possible Effect of Constraint on Product	Solution
Hardware and software platforms	Hardware/software updates over time could mean the website no longer runs properly, shows animations properly, etc. For example updates in web browsers. Software used to write the web-page could become obsolete, no longer supported. This could result in poor user experience, detriment to reputation and brand due to this, less sales overall.	Need to regularly update website to ensure compatibility with all web browsers/platforms.

Performance requirements	The budgeting app currently only works as a full screen web-page. This restricts the potential customer base, e.g. if people only want to use apps and not web pages.	Requires further CSS development so that it can be viewed on smaller screen sizes/ on mobiles, tablets, etc. Use inspect to test for accessibility and develop the HTML code to make it accessible to screen readers, etc.
Persistence storage and transactions	There are only a few seeded transaction types available to use in the drop down. This could be deemed restrictive and therefore result in poor user experience/reputation of the project. The website will only work if connected to the SQLPostgres database used for the project. Connection problems would stop transactions being saved.	Could extend the seeded data to provide additional options. Could have back-up databases. Need to maintain database.
Budgets	The website may need development, which could involve the cost of software developers, more up-to-date software, etc, which could exceed the budget available to the company. Overspend would reduce margin potential.	Need to allow for potential resource required in the annual budget.
Time	Only had one week to complete project so have constraints/limitations which could have been removed had there been more time	Spend extra time on the project to solve some of the constraints outlined above, e.g. accessibility, android layout compatibility, etc