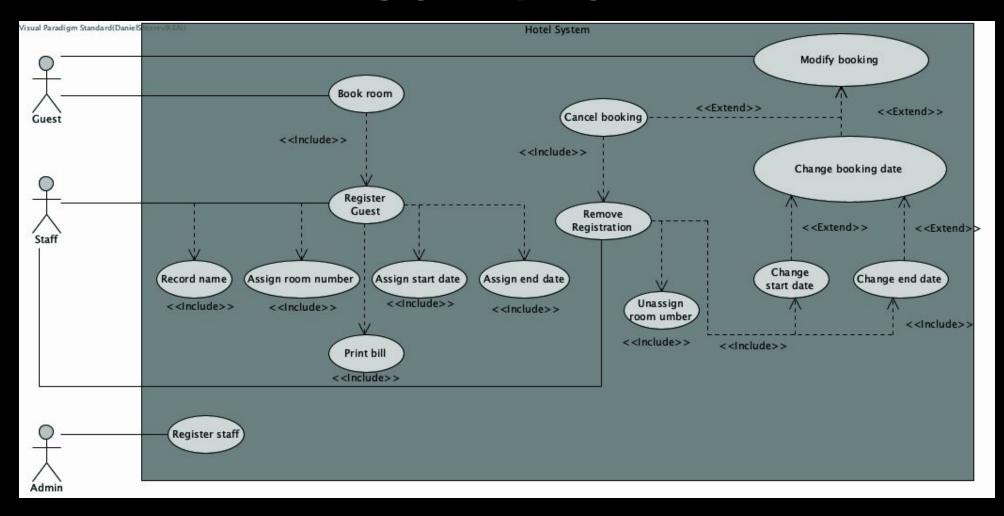


Kevin Williams Daniel Shterev Victoria Velichkova Ventsislav Ilieve

KAIZEN'S HOTEL - MODULE 3

USE CASE



USE CASE FULL DRESS

USE CASE: Hotel application Primary Actor: Staff

Stakeholders and Interests:

- Guest: Wants to book a room in the hotel

- Staff: Registers guests

- Admin: Registers staff within the program

Preconditions:

- Staff logins into program

Success Guarantee (post-conditions): Staff successfully logs into application and hotel database is loaded.

Main Success Scenario (basic flow):

1. Staff registers and logs into application.

2. Hotel application confirms login.

3. Staff registers guest to a room.

4. Hotel database is updated and saved.

5. Guest is successfully registered to a room of their choice.

6.Database is saved.

Extensions (or Alternative Flows):

*a. Staff fails to login.

1.User inputs "help" command in Hotel application

2. Staff attempts to login again.

2a. Staff login is successful.

Special Requirements:

- Hotel application has rooms hard-coded to database

- Staff members are registered

Technology and Data Variation List:

Frequency of Occurrences:

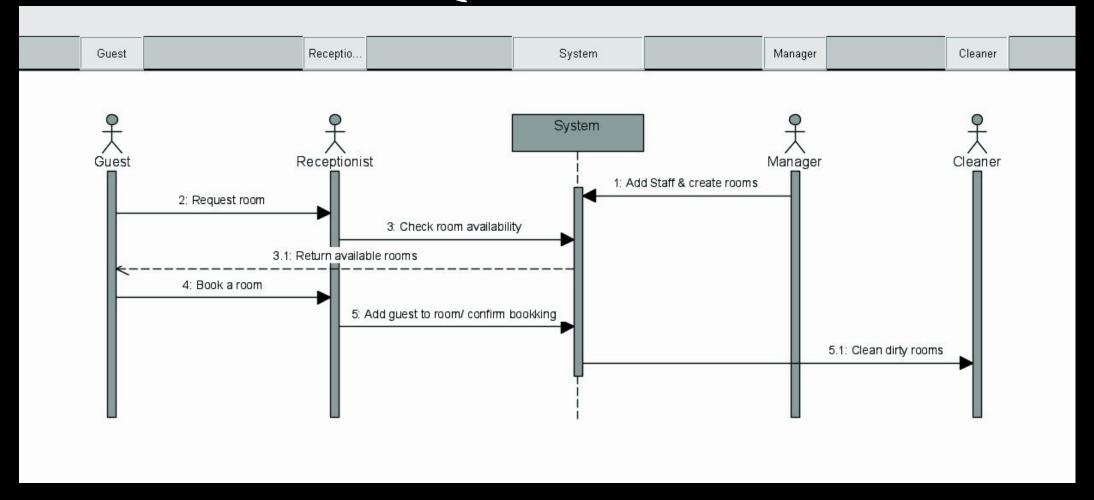
- Nearly continuous

Open Issues:

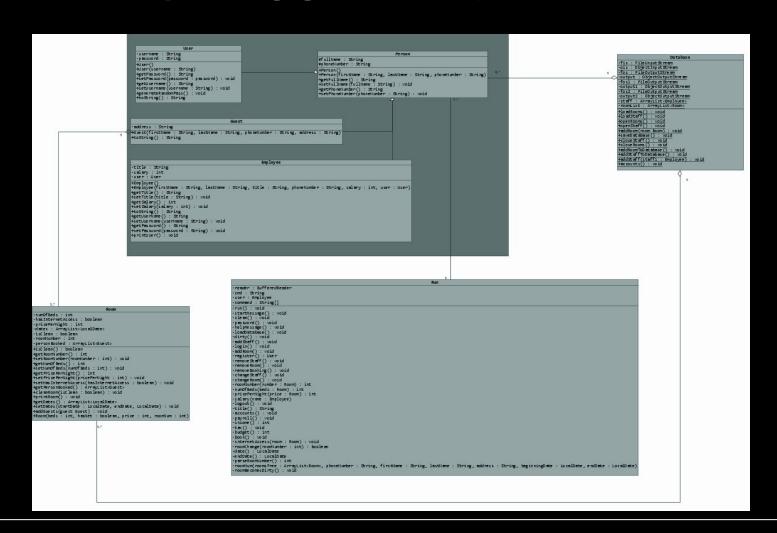
- Application maintenance

- Application secruity

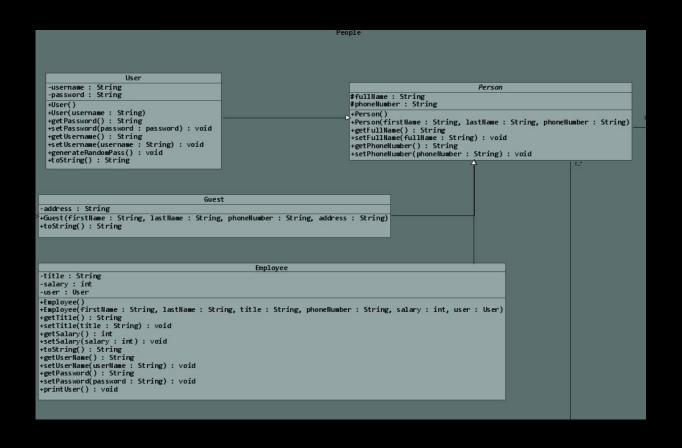
SYSTEM SEQUENCE DIAGRAM



CLASS DIAGRAM



CLASS DIAGRAM "PEOPLE"



CLASS DIAGRAM "ROOM"

```
Room
-numOfBeds : int
-hasInternetAccess : boolean
-pricePerNight : int
-dates : ArrayList<LocalDate>
-isClean : boolean
-roomNumber : int
-personBooked : ArrayList<Guest>
+isClean() : boolean
+getRoomNumber() : int
+setRoomNumber(roomNumber : int) : void
+getNumOfBeds() : int
+setNumOfBeds(numOfBeds : int) : void
+getPricePerNight() : int
+setPricePerNight(pricePerNight : int) : void
+setHasInternetAccess(hasInternetAccess : boolean) : void
+getPersonBooked() : ArrayList<Guest>
+cleanRoom(isClean : boolean) : void
+printRoom() : void
+getDates() : ArrayList<LocalDate>
+setDates(startDate : LocalDate, endDate, LocalDate) : void
+addGuests(guest Guest) : void
+Room(beds : int, hasHet : boolean, price : int, roomHum : int)
```

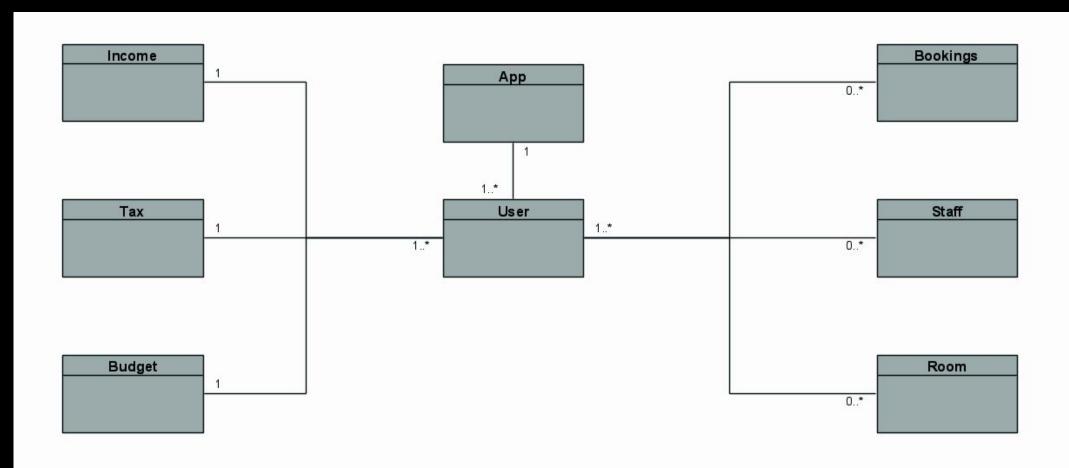
CLASS DIAGRAM "DATABASE"

```
Database
-fis : FileUnputStream
-ois : Object Input Stream
-fos : FileOutputStream
-output : ObjectOutputStream
-fos1 : FileOutputStream
-output1 : ObjectOutputStream
-fos2 : FileOutputStream
 -out put 2 : ObjectOut put Stream
-staff : ArrayList<Employee>
-roomList : ArrayList<Room>
+loadRooms(): void
+loadStaff() : void
+openRooms(): void
+openStaff(): void
+addRoom(room Room) : void
+saveDatabase(): void
+closeStaff() : void
+closeRooms() : void
+addRoomToDatabase(): void
+addSt affToDat abase(): void
+addStaff(staff1 : Employee) : void
+accounts(): void
```

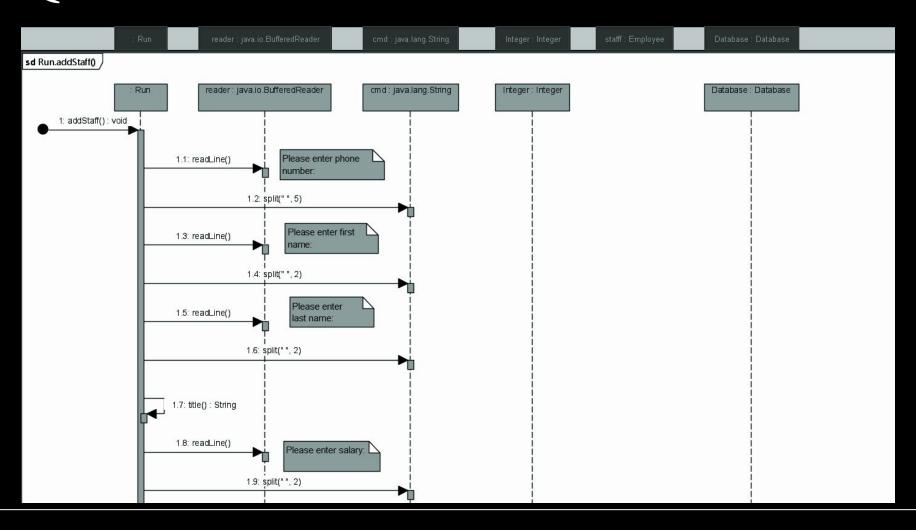
CLASS DIAGRAM "RUN"

```
reader : BufferedReader
-cmd : String
-user : Employee
 -command : String[]
 -run() : void
-startNessage() : void
-clean() : void
-pas sword(): void
 -help#essage() : void
-loadDatabase() : void
-dirty() : void
-add Staff () : void
-login() : void
 add Room(): void
 register() : User
 remove Staff(): void
 remove Room(): void
 remove Booking() : void
 -change Staff(): void
 -change Room() : void
-roomNumber(number : Room) : int
-numOfBeds(beds : Room) : int
-pricePerNight(price : Room) : int
-salary(name : Employee)
-logout() : woid
-title() : String
 -accounts() : void
 -payroll(): void
-income() : int
 -tax() : void
-budget() : int
 -book(): woid
-internetAccess(room : Room) : void
 -roomChange(roomNumber : int) : boole an
-date() : LocalDate
-endDate(): LocalDate
-parseRoomNumber() : int
-roomHum(roomsFree : ArrayList<Room), phoneHumber : String, firstHame : String, lastHame : String, address : String, beginningDate : LocalDate, endDate : LocalDate
-roomBe comesDirty() : woid
```

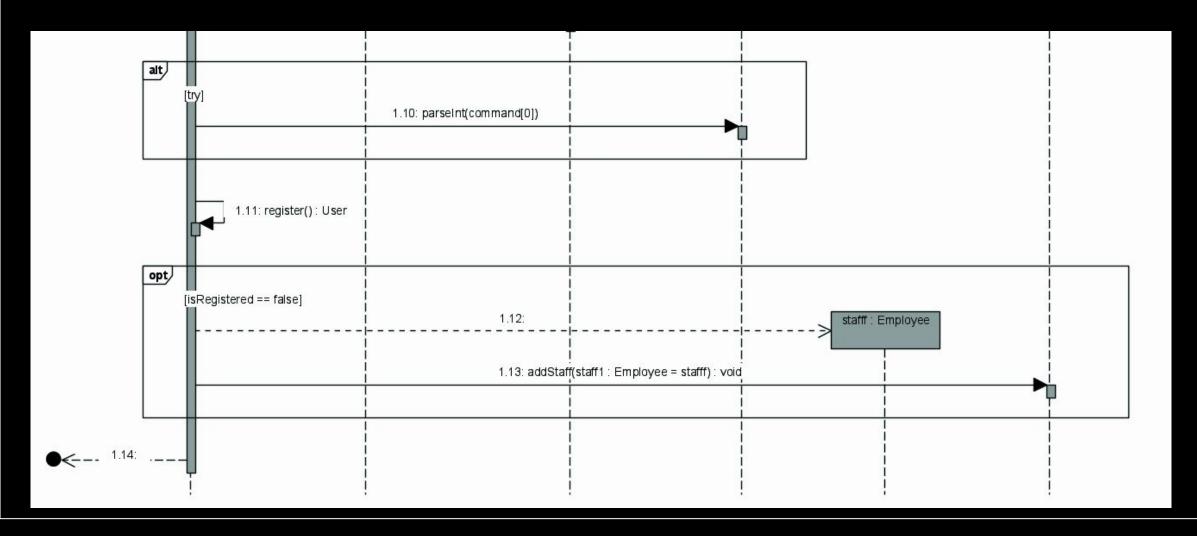
DOMAIN MODEL



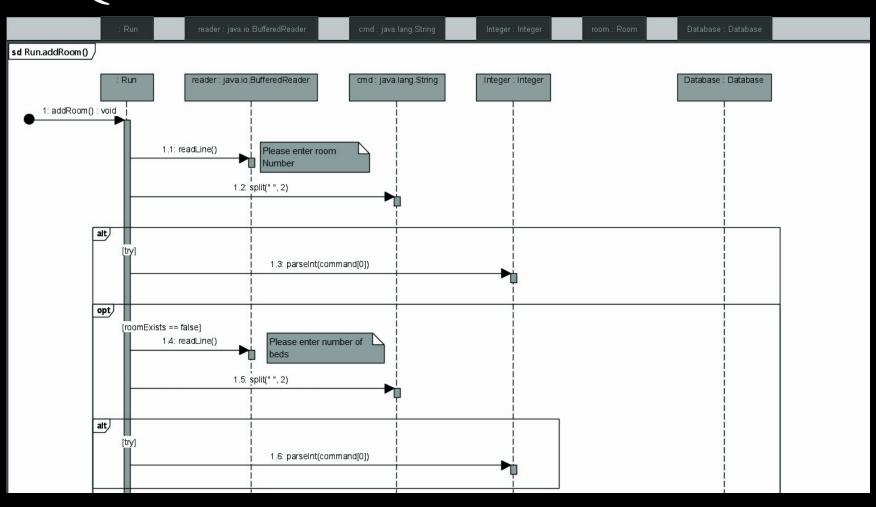
SEQUENCE DIAGRAM "addStaff"



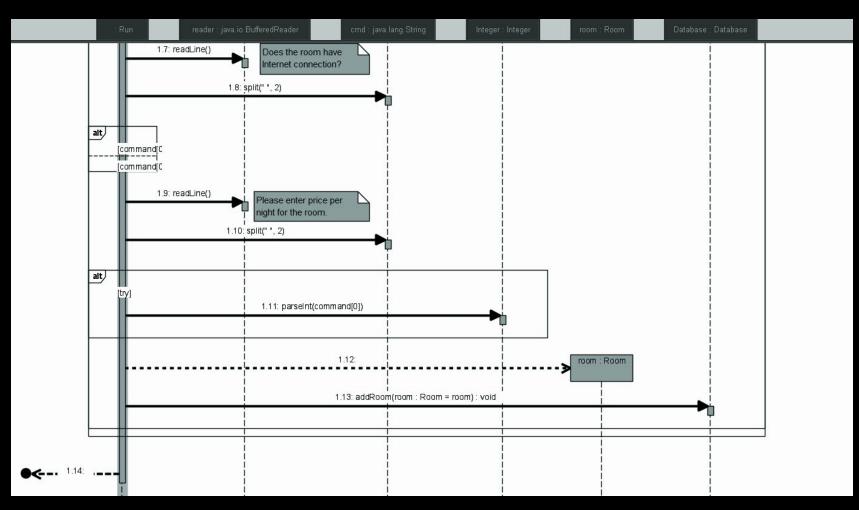
"addStaff" Continued



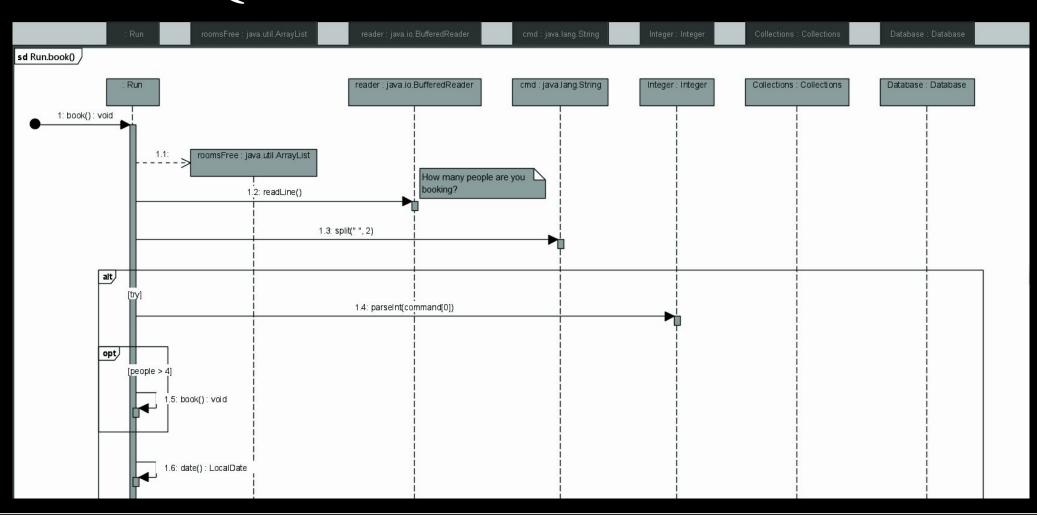
SEQUENCE DIAGRAM "addRoom"



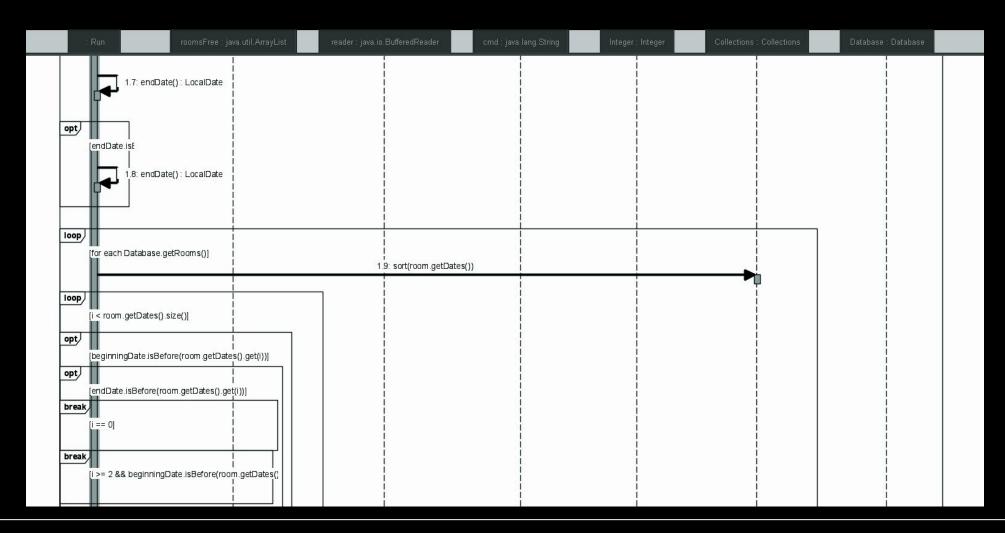
"addRoom" Continued



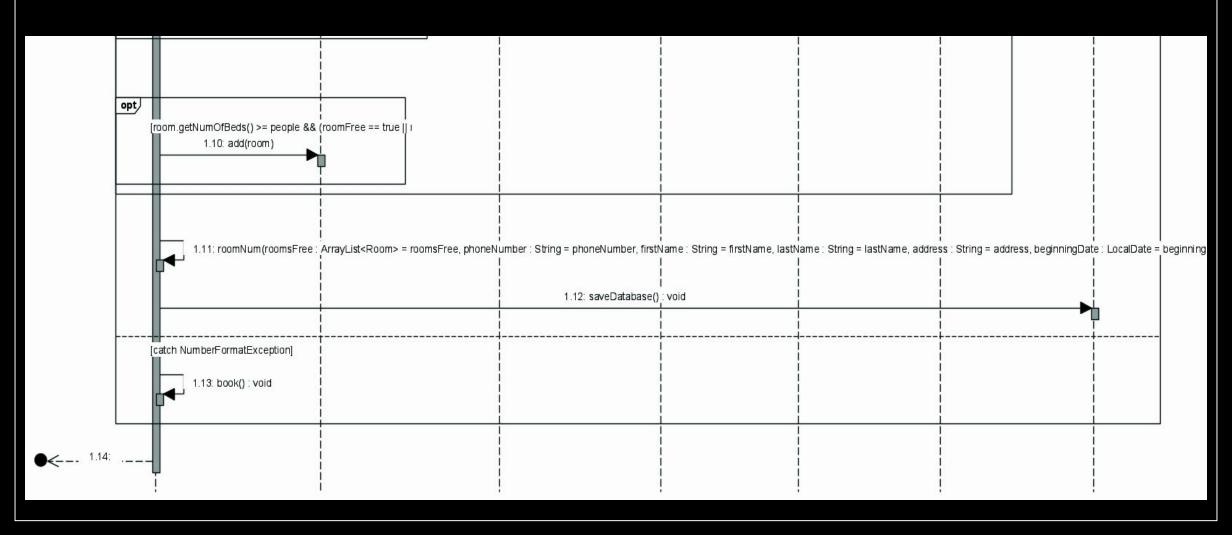
SEQUENCE DIAGRAM "book"



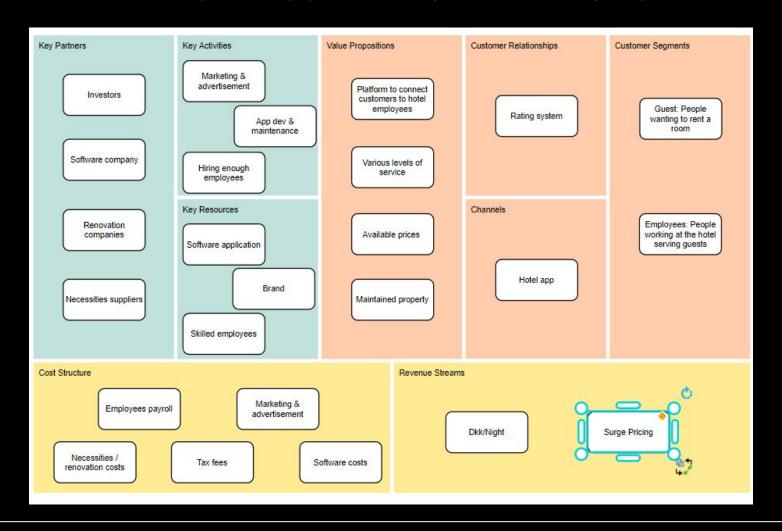
"book" Continued



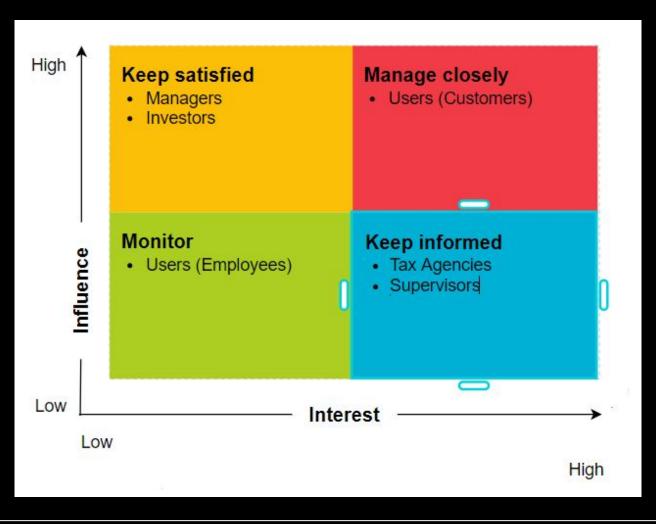
"book" Continued



Business Model Canvas



STAKEHOLDER ANALYSIS



SWOT ANALYSIS

INTERNAL

strengths

- user friendly interface
- · orderly data structure
- easy access to user info
- · safe time

weaknesses

- small customer's business
- · poor coding skills
- bad communication

POSITIVE

opportunities

- efficient booking experience
- need for hotel software

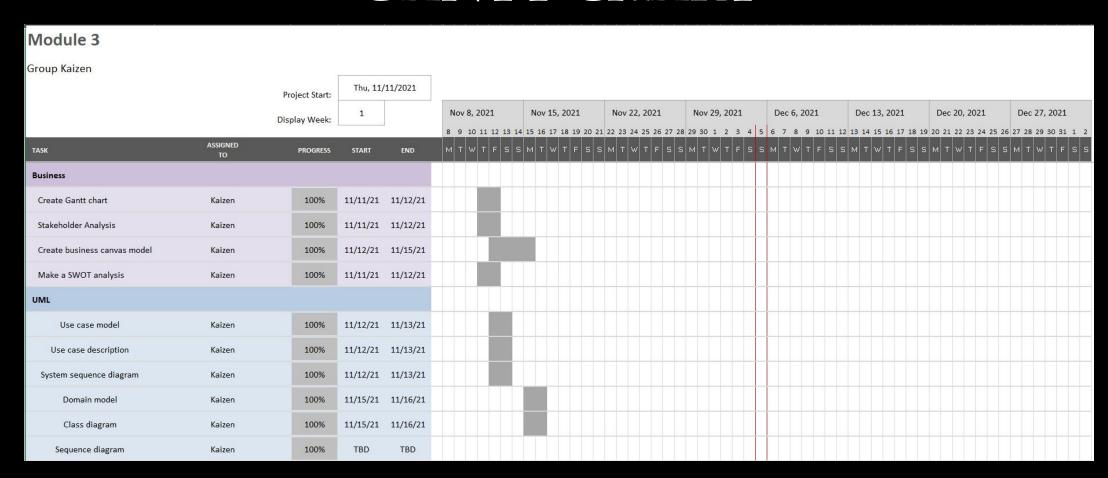
threats

NEGATIVE

- application failure
- competitors
- being declined
- security breach

EXTERNAL

GANTT CHART



GANTT CONTINUED

				Thu, 11/11/2021									
1		Project Start:	111u, 11/	11/2021				Ti and the second					
1		Display Week:	1		Nov 8, 2021	Nov 15, 2021	Nov 22, 2021	Nov 29, 2021	Dec 6, 2021	Dec 13, 2021	Dec 20, 2021	Dec 27, 2021	
					8 9 10 11 12 13 14	1 15 16 17 18 19 20 2	1 22 23 24 25 26 27 28	3 29 30 1 2 3 4 5	6 7 8 9 10 11	12 13 14 15 16 17 18 1	9 20 21 22 23 24 25 2	6 27 28 29 30 31 1 2	
		ASSIGNED TO	PROGRESS	START	END	M T W T F S S	MTWTFSS	MTWTFSS	M T W T F S S	MTWTFS	SMTWTFS	S M T W T F S	S M T W T F S S
Java Coding													
	Create Classes	TBD	100%	11/19/21	12/5/21								
	Create Methods	TBD	100%	11/19/21	12/5/21								
	Test Methods	TBD	100%	11/19/21	12/5/21								
	Combine Code	TBD	100%	11/19/21	12/5/21								
	Finalize Code	TBD	100%	11/19/21	12/5/21								
Presentation Plan													
Cre	eate Presentation (Prezi)	Kaizen	100%		12/1/21								
Assi	gn roles for presentation	Kaizen	100%		12/1/21								
	Review presentation	Kaizen	100%		12/1/21								
Insert	Insert new rows ABOVE this one												

TO THE CODE!