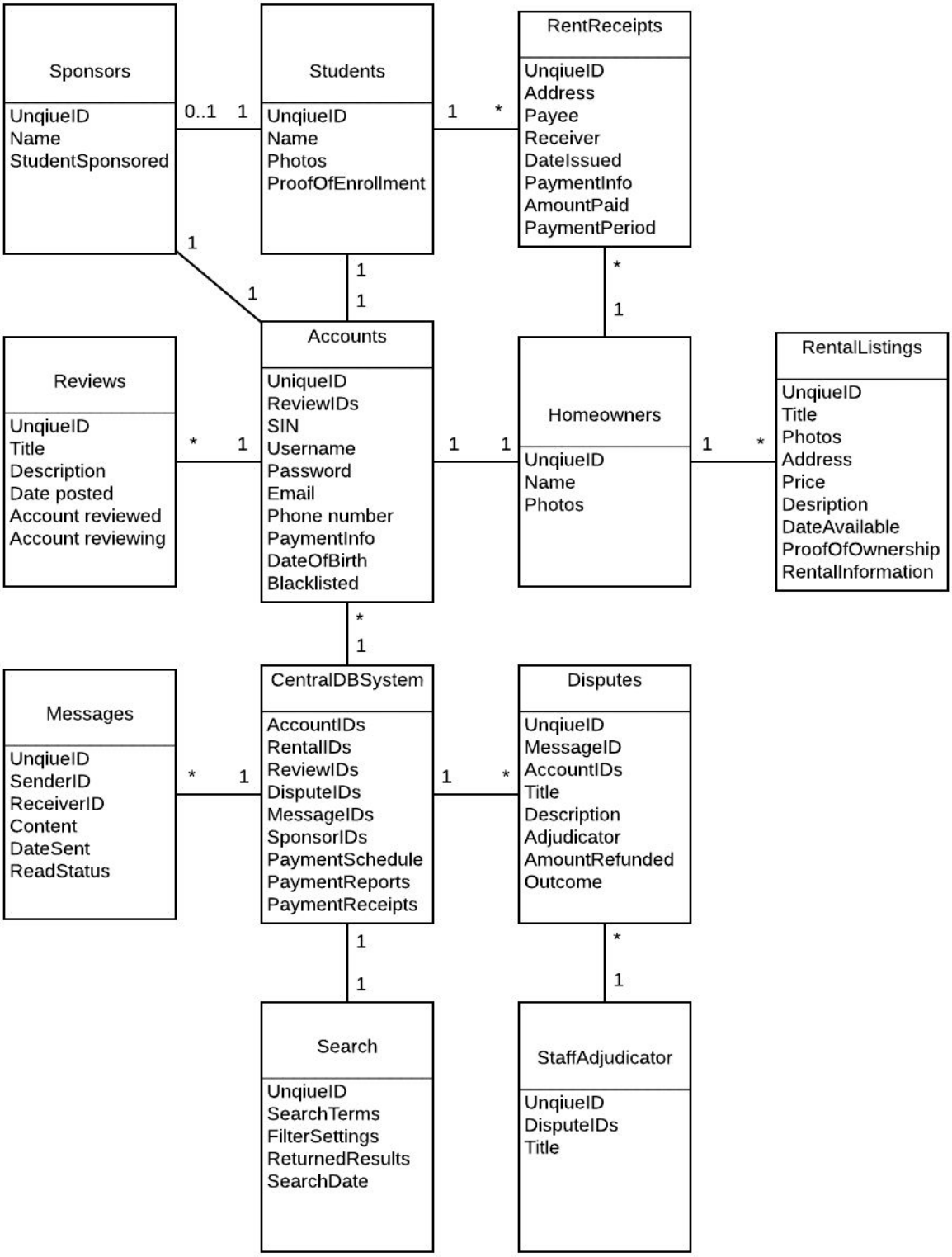
Sinking Ship Development Studios

Submitted to:
Decima Technologies

2018-11-11

Data Model


Sponsors
UnqiueID
Name
StudentSponsored
Students
UnqiueID
Name
Photos
ProofOfEnrollment
RentReceipts
UnqiueID
Address
Payee
Receiver
DateIssued
PaymentInfo
AmountPaid
PaymentPeriod
Accounts
UniqueID
ReviewIDs
SIN
Username
Password
Email
Phone number
PaymentInfo
DateOfBirth
Blacklisted
Reviews
UnqiueID
Title
Description
Date posted
Account reviewed
Account reviewing
Homeowners
UnqiueID
Name
Photos
RentalListings
UnqiueID
Title
Photos
Address
Price
Desription
DateAvailable
ProofOfOwnership
RentalInformation
CentralDBSystem
AccountIDs
RentalIDs
ReviewIDs
DisputeIDs
MessageIDs
SponsorIDs
PaymentSchedule
PaymentReports
PaymentReceipts
Messages
UnqiueID
SenderID
ReceiverID
Content
DateSent
ReadStatus
Disputes
UnqiueID
MessageID
AccountIDs
Title
Description
Adjudicator
AmountRefunded
Outcome
Search
UnqiueID
SearchTerms
FilterSettings
ReturnedResults
SearchDate
StaffAdjudicator
UnqiueID
DisputeIDs
Title
0..1
1
*

Function Points Calculations

Input Business Use Cases = 63 Function points

- BUC 1: Create a homeowner account
    - Classes Referenced = 3
    - Attributes = 13
    - 6 Function points
- BUC 2: Create a student account
    - Classes Referenced = 3
    - Attributes = 15
    - 6 Function points
- BUC 3: Create a new rental listing
    - Classes Referenced = 3
    - Attributes = 11
    - 6 Function points
- BUC 5: Rent Payment
    - Classes referenced = 4
    - Attributes = 9
    - Function points = 6
- BUC 8: Student completes apartment review
    - Classes referenced = 5
    - Attributes = 9
    - Function points = 6
- BUC 9: Student complete roommate review
    - Classes referenced = 4
    - Attributes = 10
    - Function points = 6
- BUC 10: Homeowner writes student review
    - Classes referenced = 5
    - Attributes = 9
    - Function points = 6
- BUC 11: Complaint or Dispute submitted
    - Classes referenced = 6
    - Attributes = 18
    - Function points = 6
- BUC 14: Adding sponsor to student account
    - Classes referenced = 5
    - Attributes = 7
    - Function points = 6
- BUC 16: Student rents a room from a homeowner
    - Classes referenced = 7
    - Attributes = 25
    - Function points = 6

- BUC 17: Blacklist user
    - Classes referenced = 2
    - Attributes = 3
    - Function points = 3

Output Business Use Cases = 26 Function points

- BUC 4: Display search results
    - Classes referenced = 3
    - Attributes = 7
    - Function points = 5
- BUC 12: Monthly Report
    - Classes referenced = 4
    - Attributes = 12
    - Function points = 7
- BUC 13: Annual Report
    - Classes referenced = 4
    - Attributes = 12
    - Function points = 7
- BUC 15: Student views rental location on Google maps
    - Classes referenced = 6
    - Attributes = 6
    - Function points = 7

Time-Triggered Business Use Cases = 10 Function points

- BUC 6: Pre-authorized payment setup
    - Classes referenced = 4
    - Attributes = 4
    - Function points = 4
- BUC 7: Automated Rent Payment
    - Classes referenced = 5
    - Attributes = 12
    - Function points = 6

Internally Stored Data = 66 Function points

- Accounts
    - 3 Record Elements
    - 21 Attributes
    - 10 Function points
- Rent Receipts
    - 1 Record Element
    - 8 Attributes
    - 7 Function points

- Rental Listings
    - 1 Record Element
    - 9 Attributes
    - 7 Function points
- Reviews
    - 1 Record Element
    - 10 Attributes
    - 7 Function points
- Central Database System
    - 1 Record Element
    - 9 Attributes
    - 7 Function points
- Messages
    - 1 Record Element
    - 6 Attributes
    - 7 Function points
- Search
    - 1 Record Element
    - 5 Attributes
    - 7 Function points
- Disputes
    - 1 Record Element
    - 8 Attributes
    - 7 Function points
- Staff Adjudicator
    - 1 Record Element
    - 3 Attributes
    - 7 Function points

Externally Stored Data = 0 Function points

Staff month estimate = 12.367 Staff months

Effort = (function points / 150) * function points $^{0.4}$

Input =	(63 / 150) * 63 $^{0.4}$ =	2.203 staff months
Output =	(26 / 150) * 26 $^{0.4}$ =	0.638 staff months
Time =	(10 / 150) * 10 $^{0.4}$ =	1.675 staff months
Internal =	(66 / 150) * 66 $^{0.4}$ =	2.351 staff months
External =	(0 / 150) * 0 $^{0.4}$ =	0 staff months

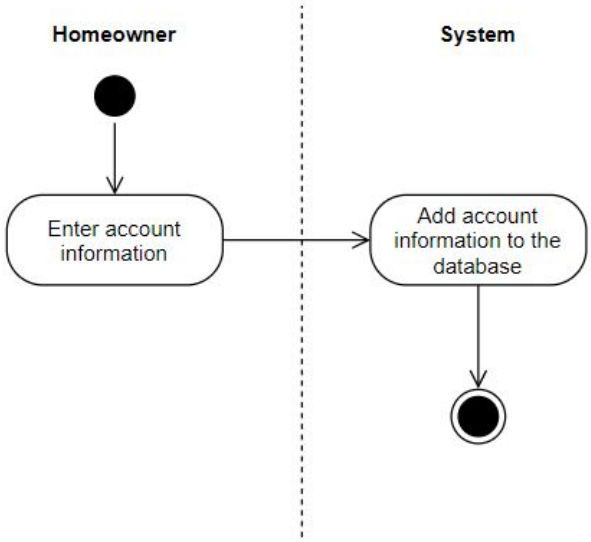
Total = 2.303 + 0.638 + 1.675 + 2.351 = 12.367 staff months

Scenarios

Business Use Case Name: Create a Homeowner account
BUC #: 1
Trigger: Homeowner information
Pre-Condition: A Homeowner has chosen to create a new account
Active Stakeholders: Homeowner (trigger), database

1. Enter required account details
2. Account information is added to the database

Outcome: Homeowner now has a homeowner account, and is able to login and access the full website.

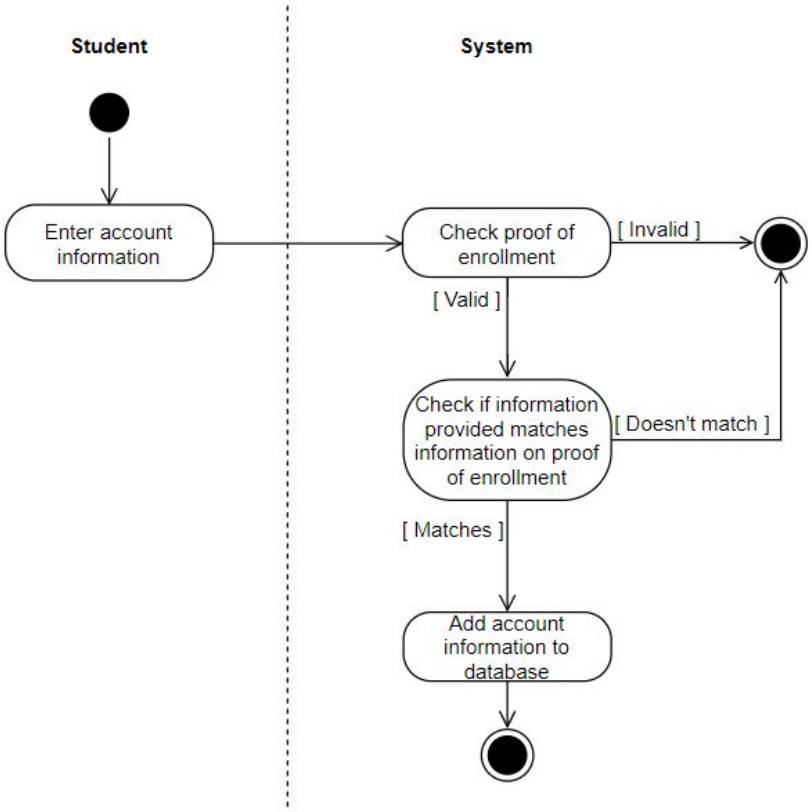


User Story #1
Description: As a homeowner I want to create an account for this service.
Rationale: I want to list my apartment/house.
Source: Discussion about BUC 1
Fit Criterion: The user provides valid information for their email address, phone number, and address. The account is successfully created.
Dependencies: email address, phone number, address
Supporting Materials: None

Business Use Case Name: Create a student account
BUC #: 2
Trigger: Student account information, proof of enrollment
Pre-condition: A student has chosen to create a new account
Active Stakeholders: student (trigger), database

1. Enter required account details, including proof of enrollment and other student information
2. Check that the information provided is valid
   2.1 Proof of enrollment must be current
   2.2 Provided information must match information on the proof of enrollment
3. Account information is added to the database

Outcome: The student now has a student account, and is able to login and access the full website.

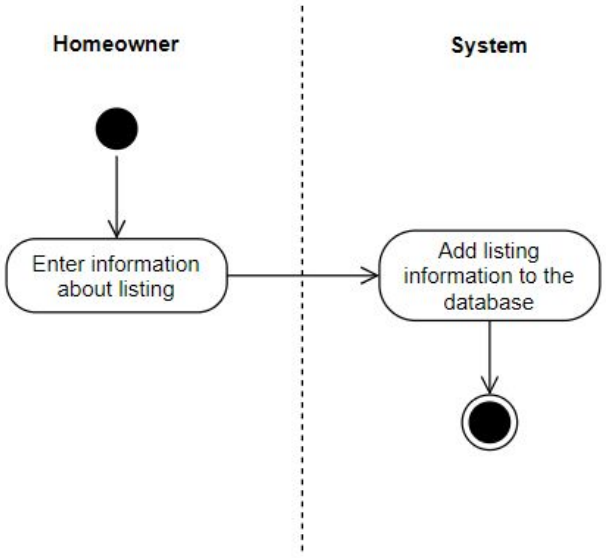


User Story #2
Description: As a student I want to create an account for this service.
Rationale: I want to rent an apartment/house.
Source: Discussion about BUC 2
Fit Criterion: The user provides valid information for their email address, phone number, address, and proof of enrollment. The account is successfully created.
Dependencies: email address, phone number, address, proof of enrollment.
Supporting Materials: None

Business Use Case Name: Create a new rental listing
BUC #: 3
Trigger: Rental information
Pre-condition: Homeowner making the listing already has an account
Active Stakeholders: homeowner (trigger), database

1. Required information is entered about the new listing
2. Listing information is added to database

Outcome: The listing is displayed as an available place to rent and shows up on the search page.

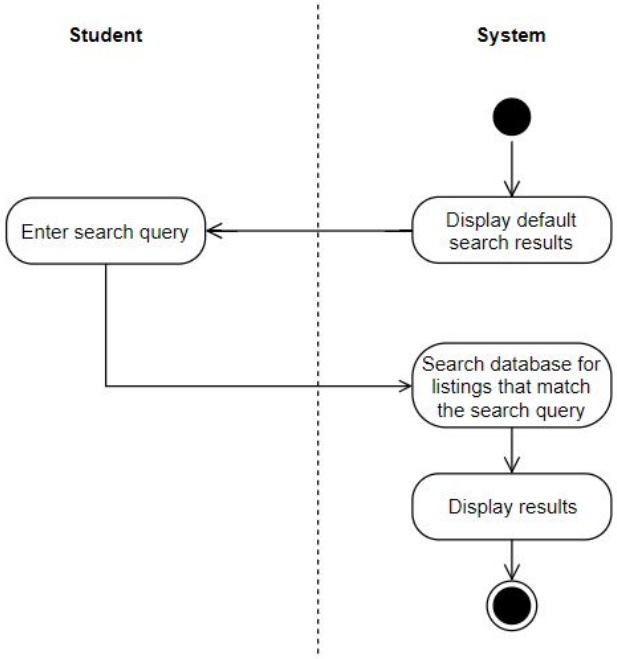


User Story #3
Description: As a homeowner I want to be able to list my apartment for students to be able to rent.
Rationale: I want students to pay rent to live in my house or apartment.
Source: Discussion about BUC 3
Fit Criterion: The apartment listing is successfully added in the system.
Dependencies: Listing information
Supporting Materials: None

Business Use Case Name: Display search results
BUC #: 4
Trigger: Students search inquiry
Pre-condition: User performing search has an account
Active Stakeholders: Student, homeowner

1. Display default search results - most recently posted listings first
2. Student enters search query, including optional filter search choices
3. Search database for listings that fulfill search requirements
4. Display relevant listings

Outcome: The student who performed the search can now see the listings that are relevant to what they want, and can choose different listings to look at from this page.

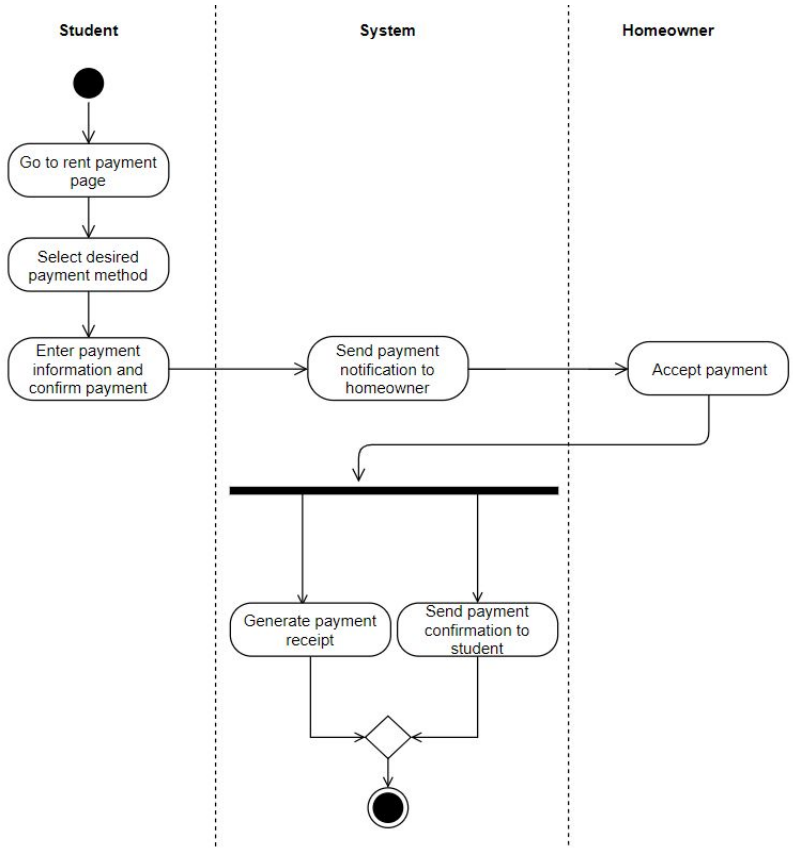


User Story #4
Description: As a student I want to to find an apartment to rent.
Rationale: I need a place to live.
Source: Discussion about BUC 4
Fit Criterion: The results are returned by the system database for the given filters. They are then shown to the user in a readable format.
Dependencies: None
Supporting Materials: Search filters

Business Use Case Name: Rent Payment
BUC #: 5
Trigger: Beginning of a new month
Pre-condition: Student is currently renting from a homeowner
Active Stakeholders: Student, homeowner

1. Open payment page
2. Select desired payment method
3. Enter payment information and confirm payment
4. Homeowner receives a notice that payment has been sent
5. Homeowner accepts the payment
6. Send confirmation to the student that their payment has been received
7. Generate a receipt and send it to the student

Outcome: Required rent payment for that month is now paid

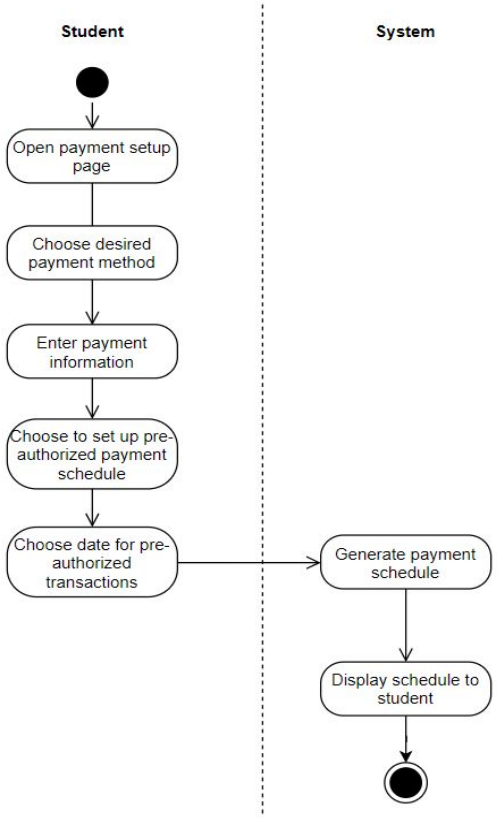


User Story #5
Description: As a student and a student I need to be able to pay my rent.
Rationale: I need to pay rent to avoid having a dispute filed against me.
Source: Discussion about BUC 5
Fit Criterion: The payment system tells us that payment has successfully been received. The system tells the user that their payment has been accepted.
Dependencies: Student needs to be actively renting a listing, payment information.
Supporting Materials: None

Business Use Case Name: Pre-authorized payment setup
BUC #: 6
Trigger: Student decides to set up this feature offered by the system
Pre-condition: Student is currently renting from a homeowner
Active Stakeholders: Student (trigger), Associated Bank

1. Open payment page
2. Select desired payment method
3. Enter payment information
4. Choose to set up a pre-authorized payment schedule
5. Choose date for pre-authorized payments to go through
6. Generate payment schedule and display it to the student

Outcome: Pre-Authorized Payment Set up complete. Students rent is now automatically paid based off payment schedule.

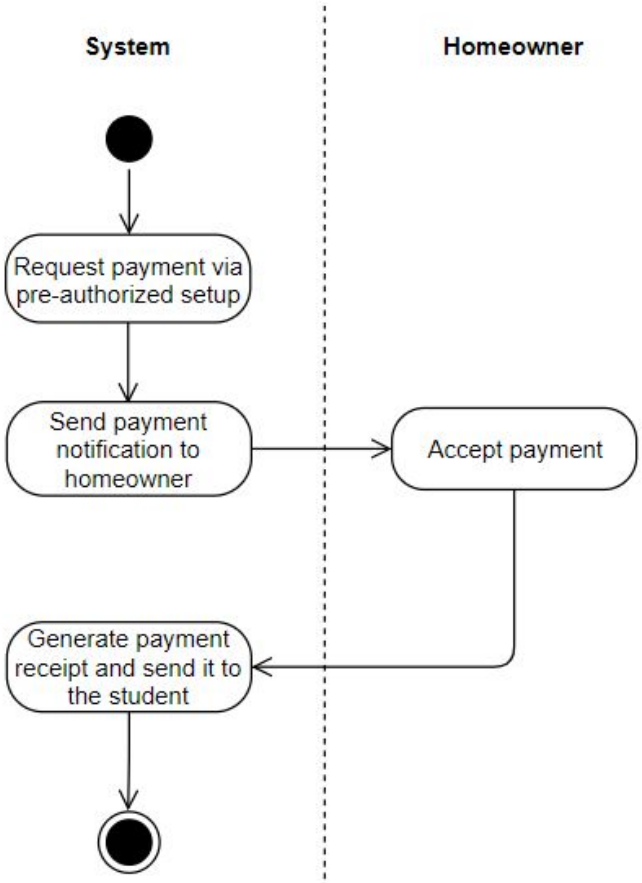


User Story #6
Description: As a student I want to be able to set up a pre-authorized payment to automatically pay rent each month.
Rationale: I want the ease-of-use associated with not having to log in to the system each month to manually pay rent.
Source: Discussion about BUC 6
Fit Criterion: The pre-authorized payment is approved by the user's bank.
Dependencies: Bank account.
Supporting Materials: None

Business Use Case Name: Automated Rent Payment
BUC #: 7
Trigger: Time triggered, (When the beginning of a new month starts)
Pre-condition: Must have pre - authorized payments set up
Active Stakeholders: Student, Associated Bank

1. Students bank information is saved to system
2. Bank automatically submits Students payment for rent when scheduled time is reached
3. Payment receipt confirmation sent via email to both Student and homeowner

Outcome: Students rent is now paid for using pre-authorized payment

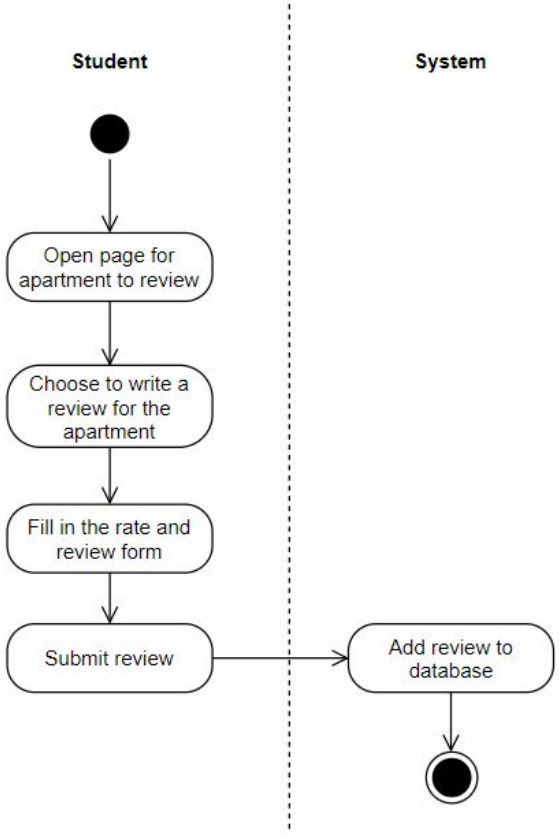


User Story #7
Description: An automated pre-authorized payment is processed by our system.
Source: Discussion about BUC 7
Fit Criterion: The pre-authorized payment is approved by the users bank.
Dependencies: BUC 6, having a pre-authorized payment set-up
Supporting Materials: None

Business Use Case Name: Student completes apartment review
BUC #: 8
Trigger: Student chooses to review apartment
Pre-condition: Student has lived in apartment being reviewed
Active Stakeholders: Student (trigger), homeowner

1. Student chooses apartment to rate and review
2. Rating is selected by student
3. Student includes review of apartment to help explain chosen rating
4. Rating and review are added to database

Outcome: Student rating and review of apartment is now posted on the apartment's page.

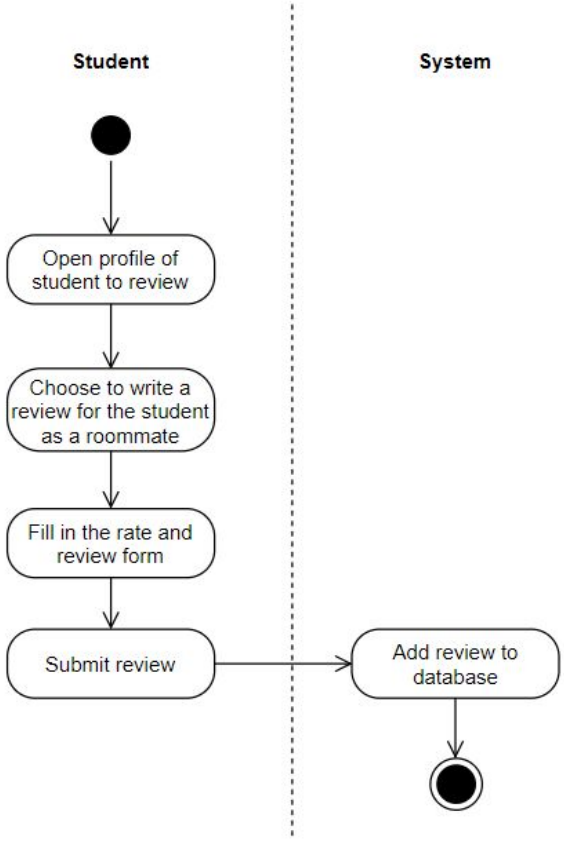


User Story #8
Description: As a student I would like to be able to rate and review my apartment.
Rationale: Giving feedback on apartment
Source: Discussion about BUC 8
Fit Criterion: Review is successfully added to listing page.
Dependencies: Review details.
Supporting Materials: None

Business Use Case Name: Student completes roommate review
BUC #: 9
Trigger: Student chooses to review roommate
Pre-condition: Student has to have lived with roommate before
Active Stakeholders: Student (trigger), roommate

1. Student chooses roommate to rate and review
2. Student rates and writes review for roommate
3. Review is saved to database and posted to roommate profile

Outcome: Student rating and review of roommate is posted on roommates profile and added to the systems database.

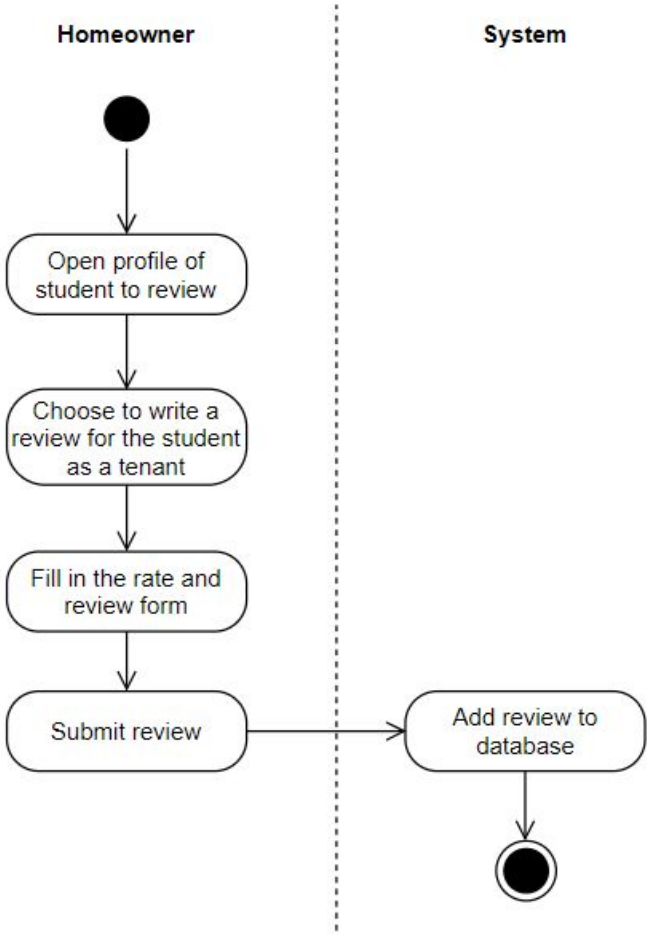


User Story #9
Description: As a student I would like to be able to rate and review my roommates.
Rationale: Giving feedback
Source: Discussion about BUC 9
Fit Criterion: Review is successfully added to roommate profile.
Dependencies: Review description.
Supporting Materials: None

Business Use Case Name: Homeowner writes student review
BUC #: 10
Trigger: Homeowners chooses to review student
Pre-condition: Individual being reviewed must be a student.
Active Stakeholders: Homeowner(trigger), student

1. Homeowner selects student to review
2. Homeowner writes and submits review
3. Review is added to the database

Outcome: The Homeowners review is posted to the students profile.

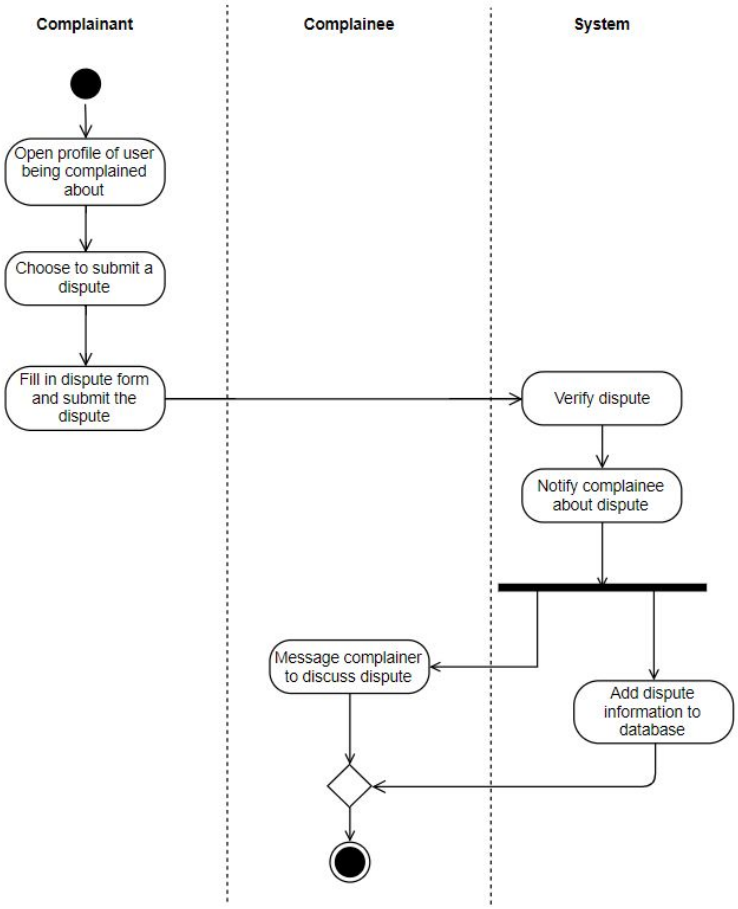


User Story #10
Description: As a homeowner I would like to be able to write reviews about students that are renting from me.
Rationale: Giving feedback on students that are renting
Source: Discussion about BUC 10
Fit Criterion: Review is successfully added to student profile.
Dependencies: Student is renting from landlord, review description.
Supporting Materials: None

Business Use Case Name: Complaint or Dispute submitted
BUC #: 11
Trigger: Student or homeowner is bothered by something resulting in a complaint being filed
Pre-condition: To file a complaint or dispute, you must be either a student or homeowner of an apartment.
Active Stakeholders: Student or homeowner(trigger),

1. Student or homeowner opens page of individual being complained about
2. User chooses to make a dispute against the chosen user
3. Dispute is written and submitted
4. Dispute is verified by someone working for the company
5. Both parties involved in the dispute contact each other through the messaging system

Outcome: A complaint has be filed and will be stored in the systems database.

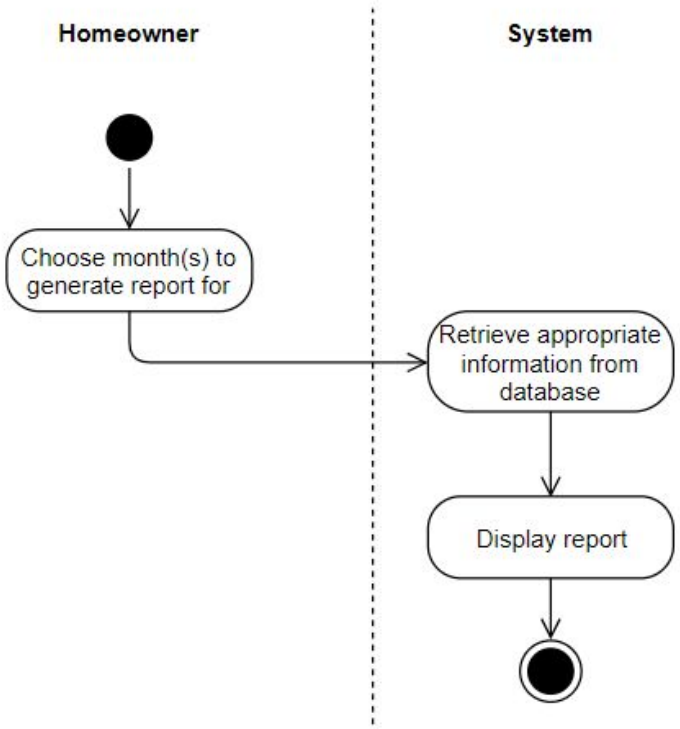


User Story #11
Description: As a student or homeowner I want to submit a dispute.
Rationale: I have a dispute to file upon a homeowner or student.
Source: Discussion about BUC 11
Fit Criterion: Dispute is successfully filed in our system.
Dependencies: Dispute information.
Supporting Materials: None

Business Use Case Name: Monthly Report
BUC #: 12
Trigger: Homeowner requests a report
Pre-condition: Homeowner chooses to receive monthly report before the month ends.
Active Stakeholders: Homeowner, Developers

1. Homeowner chooses the month(s) they want a report for
2. Retrieve information from database
3. Generate report using retrieved information
4. Display report

Outcome: A detailed report is sent displayed to the homeowner showing the details

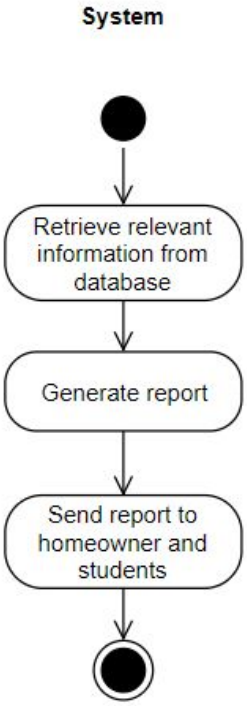


User Story #12
Description: As a student or homeowner I would like to receive a report on monthly payments on a specific date each month.
Rationale: I want to keep a history of payments.
Source: Discussion about BUC 12
Fit Criterion: Report is generated and sent successfully.
Dependencies: User must be a student renting a house or a landlord renting out their house.
Supporting Materials: None

Business Use Case Name: Annual Report
BUC #: 13
Trigger: Occurs at the end of the year
Pre-condition: Must be a Homeowner or student to receive
Active Stakeholders: Developers, Homeowner, Student

1. Information is fetched from the database at the end of the term
2. End of term report is automatically generated
3. Report sent to Homeowner and Student

Outcome: Report is sent to Homeowner and Students at the end of the term.

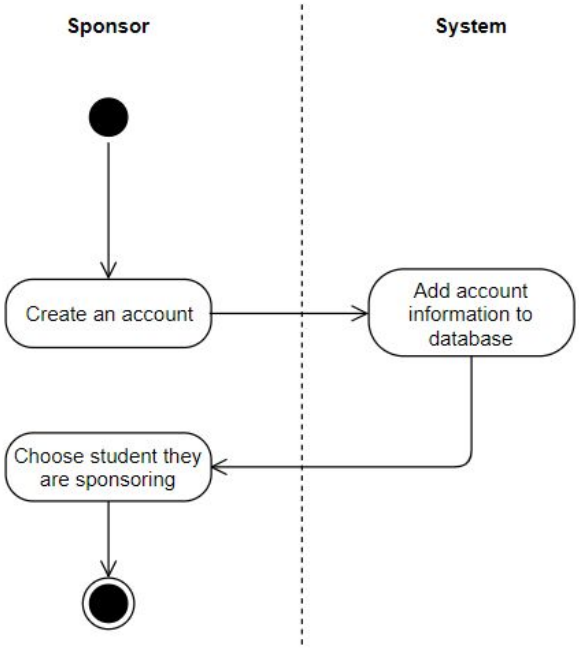


User Story #13
Description: As a homeowner I would like to receive a annual report on all payments made by students.
Source: Discussion about BUC 13
Fit Criterion: Report is generated and sent successfully.
Dependencies: User must be a landlord renting out their house to students.
Supporting Materials: None

Business Use Case Name: Adding Sponsor to student account
BUC #: 14
Trigger: Student has sponsor to rent for them
Pre-condition: Students must be living in an apartment
Active Stakeholders: Student (trigger), Sponsor

1. A new account is created for the sponsor
2. Sponsor information is collected
3. Sponsor indicates which student they are sponsoring.

Outcome: Sponsor is now added to the students account and sponsor pays rent on students behalf.

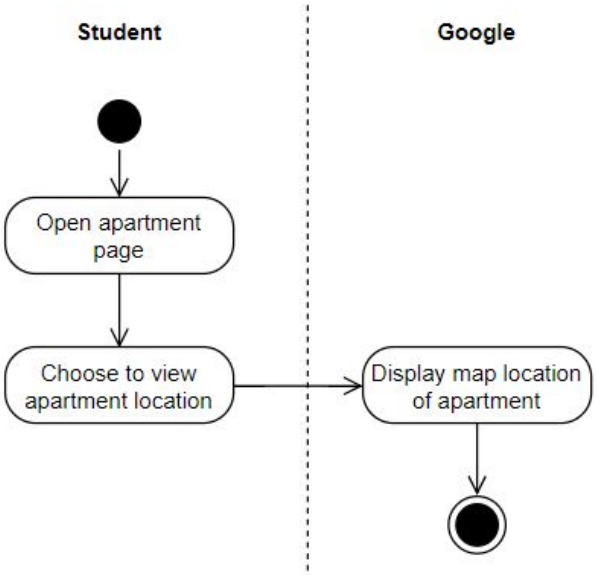


User Story #14
Description: As a student I would like to add a sponsor to my account to help me pay my rent.
Rationale: I cannot afford rent on my own.
Source: Discussion about BUC 14
Fit Criterion: Sponsor is linked to student account as a payer.
Dependencies: None
Supporting Materials: None

Business Use Case Name: Student views rental location on Google Maps
BUC #: 15
Trigger: Student activates google maps location feature
Pre-condition: Student must have an account on the system
Active Stakeholders: Student (trigger),

1. Student opens the page of a rental listing
2. Student selects option to view the location in Google Maps
3. Google Maps is opened and location is displayed

Outcome: Location of rental listing is open in Google Maps for student to look at.

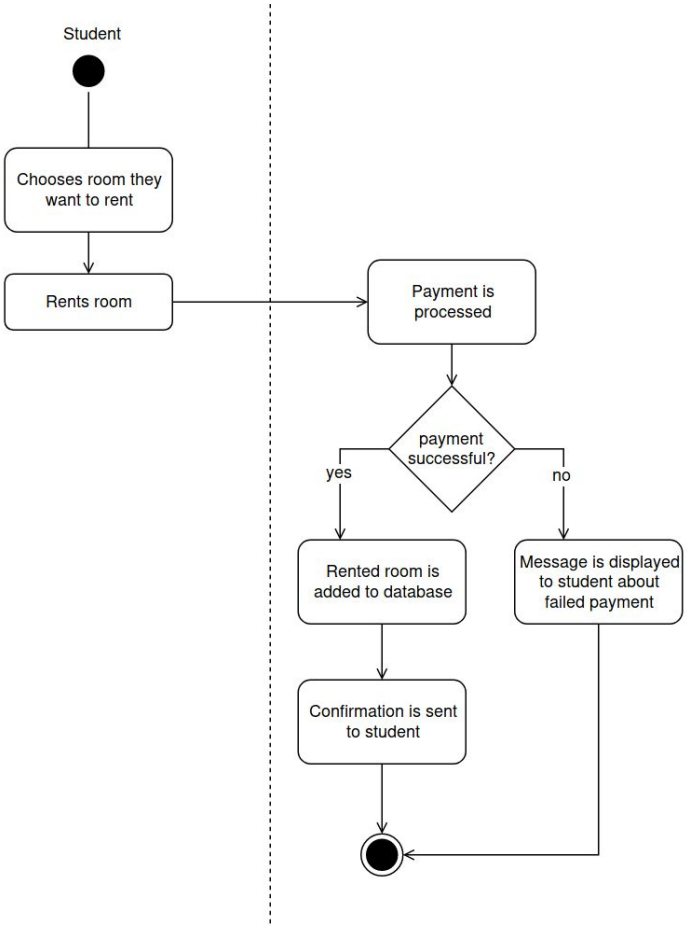


User Story #15
Description: As a student I would like to be able to see the location of the apartment or house that I looking to rent.
Rationale: I want to make sure that the house is in a location that I like.
Source: Discussion about BUC 15
Fit Criterion: Map is displayed to user with the correct location of said apartment or house.
Dependencies: None
Supporting Materials: Apartment/house listing

Business Use Case: Student rents room
BUC #: 16
Trigger: Student decides to rent room from homeowner
Pre-condition: Student must have an account on the system
Active Stakeholders: Student (trigger), Homeowner

1. Student searches for apartment to rent
2. Student decides which apartment they would like to rent
3. Decision is made between student and homeowner to let student rent room
4. Payment is made

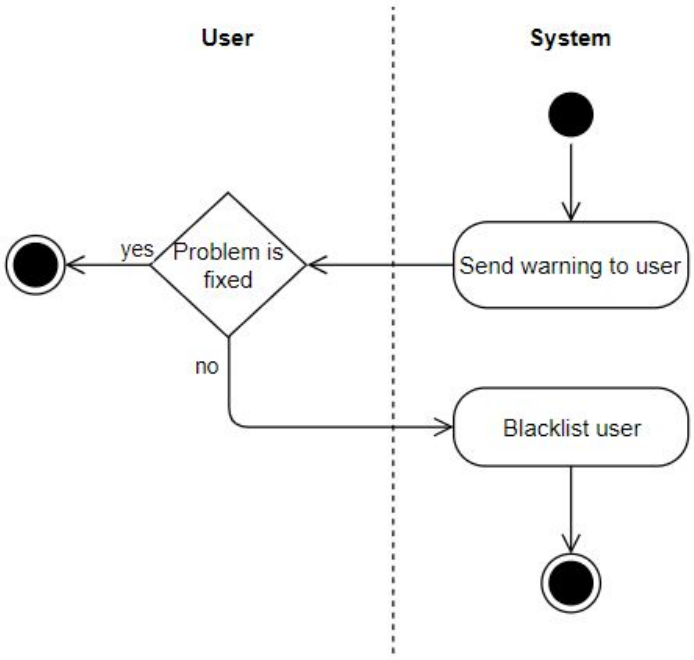
Outcome: Student is now renting room from Homeowner



User Story #16
Description: As a student I would like to be able to rent a room in an apartment.
Rationale: I like the apartment and would like to live there.
Source: Discussion about BUC 16
Fit Criterion: Room is rented and confirmation has been successfully sent to student.
Dependencies: Payment information
Supporting Materials: Apartment/house listing

Business Use Case: Blacklist user
BUC #: 17
Trigger: Multiple disputes have been filed against the user
Pre-condition: User must have an account
Active Stakeholders: User being blacklisted

1. User receives a warning that they will be blacklisted
2. User is given a period of time to fix the problem
3. Blacklist user if problem has not been fixed
   A3.1 If problem has been fixed, notify user that they will not be blacklisted



User Story #17
Description: As a homeowner I have been blacklisted from creating listings or as a student I have been blacklisted from renting from homeowners.
Rationale: Multiple disputes have been filed against me and I have failed to respond accordingly.
Source: Discussion about BUC 17
Fit Criterion: User cannot use the system any longer.
Dependencies: Multiple disputes filed against user.
Supporting Materials: None