LICS Assignment description

Assumptions:

1) Assuming that once the user balance is over then, he can't add more money into his account, since we have not learnt to take input in UPPAAL yet.
2) Assuming that the user would either enter a correct email or incorrect email, depending on his choice, but not actually entering email in model since we cannot take input yet and also authentication cannot be done by the system because of no existing database with it, hence I have authenticated user on the basis of his choice of giving correct details or not.
3) Assuming that one can view his/her membership details/history on clicking the corresponding option since we haven't learnt to show some output on UPPAAL.
4) Assuming that during payment the user give correct details of Credit/Debit card, again because we haven't learnt to give input in UPPAAL.
5) Assuming that User won't be logged out after every unsuccessful transaction, while he will be given an option to log out from the OptionsPanel itself.

Properties Verifying:

1) A[] (not(Website.SelectingMembership and Website.MembershipHistory)) :
   This formula checks that for all paths, SelectingMembership and MembershipHistory states in Website can't be together true, which ensures safety.
2) A[] (not(Website.PaymentFailed and Website.PaymentSuccessful)) :
   This formula checks that for all paths, PaymentFailed and PaymentSuccessful states in Website can't be together true, which ensures safety.
3) A<>(Website.TransactionInitiated imply Website.Ready) :
   This formula checks that for all paths, sometime in future TransactionInitiated would result in (or imply to) Ready state(initial state).
4) A<>(Website.SelectedGold imply Bank.BalanceCheckG) :
   This formula checks that for all paths, sometime in future Website.SelectedGold would result in (or imply to) Bank.BalanceCheckG state(initial state).
5) A[] (not deadlock) :
   This formula checks that our model don't do in deadlock.

Description:

The model have 3 entities as our system namely bank, user, website. The user actively interacts with the website in order to purchase gold membership or platinum membership, while the bank checks if the user have enough money to do the transaction or not, if he has enough money, then the cost of the membership gets deducted from the users account.

First of all, the website asks users his details to continue, user have to give correct email and correct password to make bool login to true and to continue to the OptionsPanel in the website where he/she will be given 3 options, View membership details, Select membership type, or LogOut, if user tries to view membership details/History then he will be given option to go back only, assuming that he could see the membership details/history. If the user tries to logout, he will be logged out and sent to the initial state and login bool would be changed to false. Then if he tries to Buy Membership card then he has 2 options:-

1) Buy Gold Membership which makes Bool BGold true,
2) Buy Platinum Membership which make Bool BPlat true.

Both of the options proceeds to ask the person to initiate the transaction, after which the website refers to the bank to actually check if the user have enough money to carry out the transaction, the bank first check which Bool is true, either BGold or BPlat, then compares the users balance with the price of the selected Membership card, and activates the one of the channel from GoldOK, PlatOK, GoldNOK,PlatNOK, which determines if the card can be bought or not, and subsequently deducting the amount from the user's bank account, activation of these channels also changes state in website to either PaymentSuccessful (if GoldOK or PlatOK activated) or PaymentFailed(if GoldNOK or PlatNOK activated). From PaymentSuccessful state the website Informs user that the payment is done through PaymentDone channel, which Logs out the user from the website doing logIn bool to false and taking both user and website to initial state, while from PaymentFailed state, the website informs the user that the Payment is not done through the PaymentNotDone channel, which takes the user again to the OptionsPanel state where he may wish to try another transaction or log out of the website.