Creating a Workflow System using Django

CE219 Assignment

Evaldas Senavaitis (1402039)

1. Elicitation of System Details

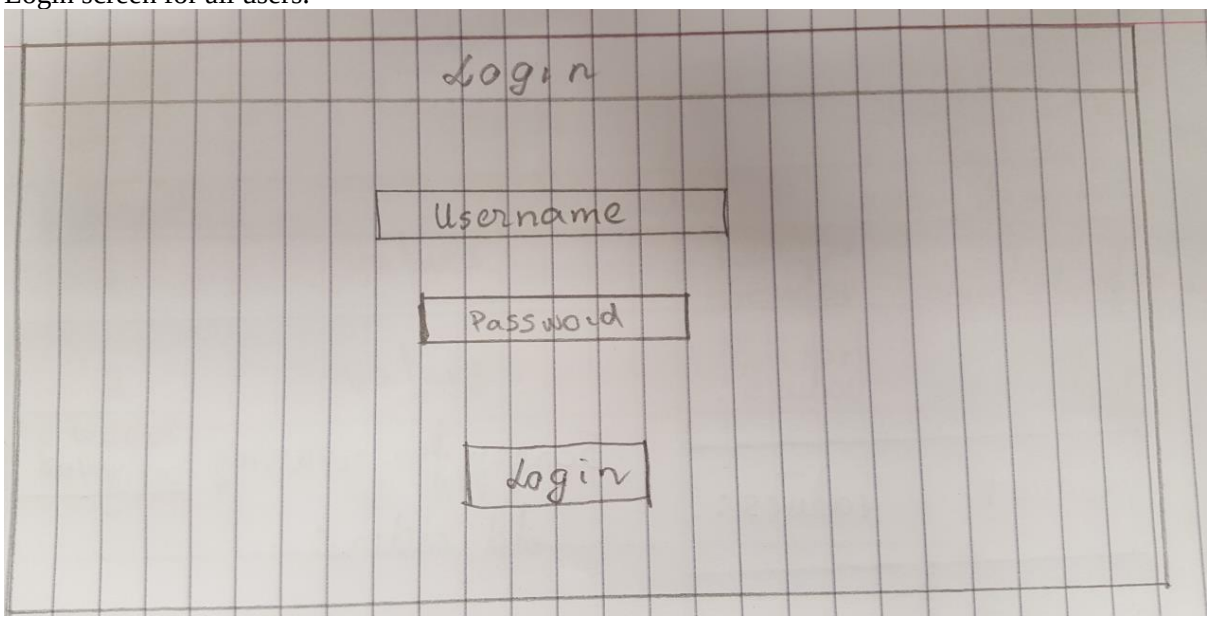
Describe what procurement system does.

Procurement is a accounting system, that keeps track of money "in and out", including budget and overspending. It focuses on a making that you spend your money on the things you need.

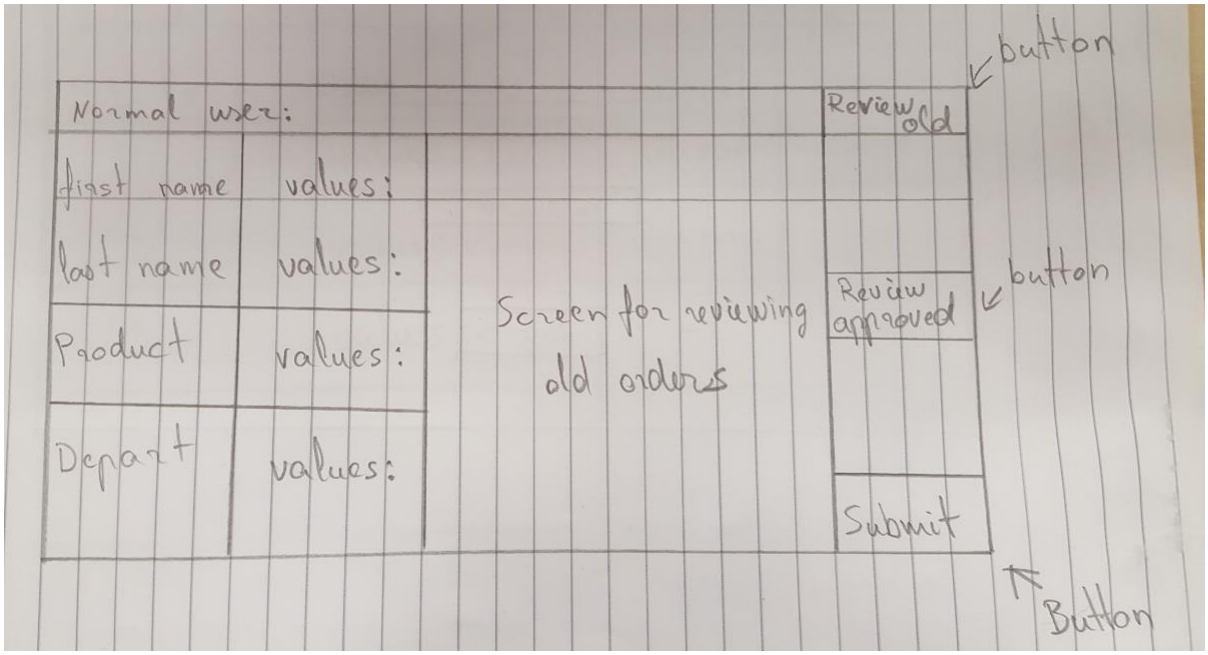
Login to the system: Username, Password. System has a approval staff that keep orders in form, covering them with diferrent approval levels. The approval system consist of: Requisitioners, Workflow, Approvler, Purchase order.
Requisitioner is a person who wants to buy something, Workflow is a authorisation path of request, Approvler is a special kind of user that authorises or rejects orders, Purchase order is a final step at which supplier ships out its goods. Goods can be send back in 7 days period. Before even goods have been delivered, supplier gives you an invoice of your order for 100% approval.
One thing is that requisition must be above 0 £ price, as you cannot receive goods for free.

2. Specification of System

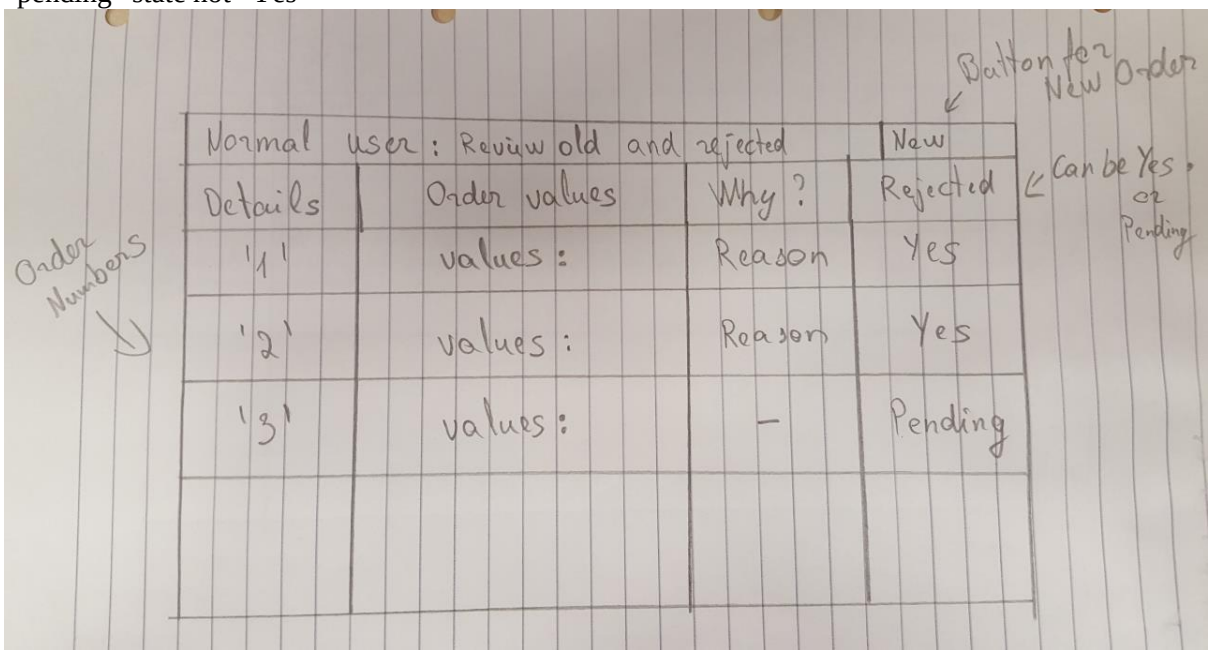
Login screen for all users:



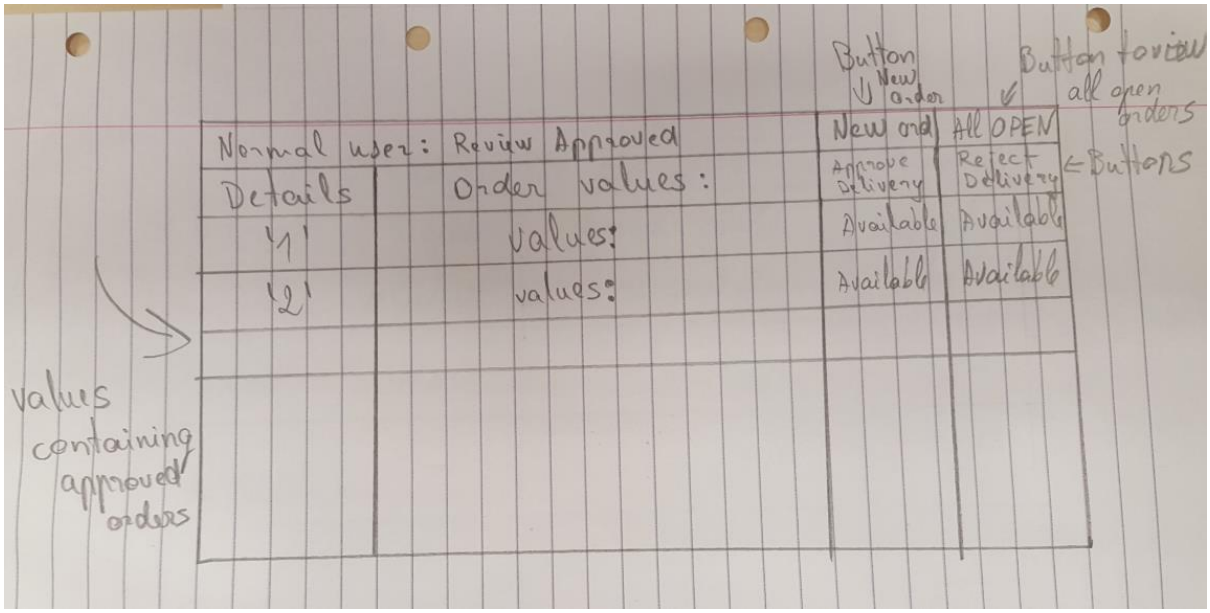
Main screen of the normal user, he requires to input first and last names ir products and department name for his new purchase order in values: spaces. There are buttons for Review old and rejected orders, Review Approved and Submit button or new order.



Screen for Review old and reject order. At the top there is New button for creating new order, below listing all rejected and old orders with their values in values: space with a reason why. Why can only be with value only if reject section says "YES", if order is not rejected it is seen as old one and in "pending" state not "Yes"



Screen for normal user: Review Approved. It shows all approved orders with all their order numbers details in values: space. At the top there are two buttons: New order, ALL OPEN. ALL OPEN is for viewing all open orders. Next user can select to approve or reject delivery "Available" being as buttons for user easy interface.



First level user will use the same screen as a normal user for login, from there he will be passed to his main screen in which he can see all open purchase orders that need first level approval. He can review the those orders and decide if he wants to approve them or not.

Second level user as well will use the same login screen as other users, but he will have more extent main screen, in which he can see first level user approved purchase orders but he will have an option how much he want to see on that screen, ex: he select to see more than 5 and it will display only if there are 5 or more first level approved orders, to addition to that he can review them just like first level user and can approve them or reject them.

Finance Director uses the same login screen, to his main screen can display all open purchase orders or look at all closed purchase orders that did not get approved. Sorting selection is available to him for more neater look and greater navigation between orders.

3. Implementation

3.1 Example Level Two

In the report, outline how you implemented the system, what problems you had and how you solved them. Include screen shots of the various forms in your system. Include a discussion on the interface with the MongoDB database.

I implemented system rather easily, because I encountered numerous problems which cannot be implemented, more on those later. From a provided template I added some code to the view.py in the workflow directory.
First thing I did to view.py I added some to to raise_req() method, like getting number from the user he inputs, then added a insert variable which hold a insert_one() method, it holds all the details user inputs and some pre-defined ones as well. Finishing raise_req() method witch a "Go to authorisation page: " Http response which allows user to jump tp authorisation or in other words approval screen.

Only a small detail was added to requisition() and it is a number input box in http response. A bigger part was to produce progress() method which now holds mongodb and can only be accessed by issuing a secure token from database which is a purchase number user inputs. Progress() shows you that purchase order was approved and to addition to that displays whole mongodb for reference. Whole mongodb is stored on a list for users to look, and it is not in a neat formatted way as that is not possible with current software packages.

Moving to authorise() method it as well holds a mongodb for method to be able to access it and stored in a list. This method only asks you to input a number of purchase order which can be seen in displayed mongodb below and nothing more.

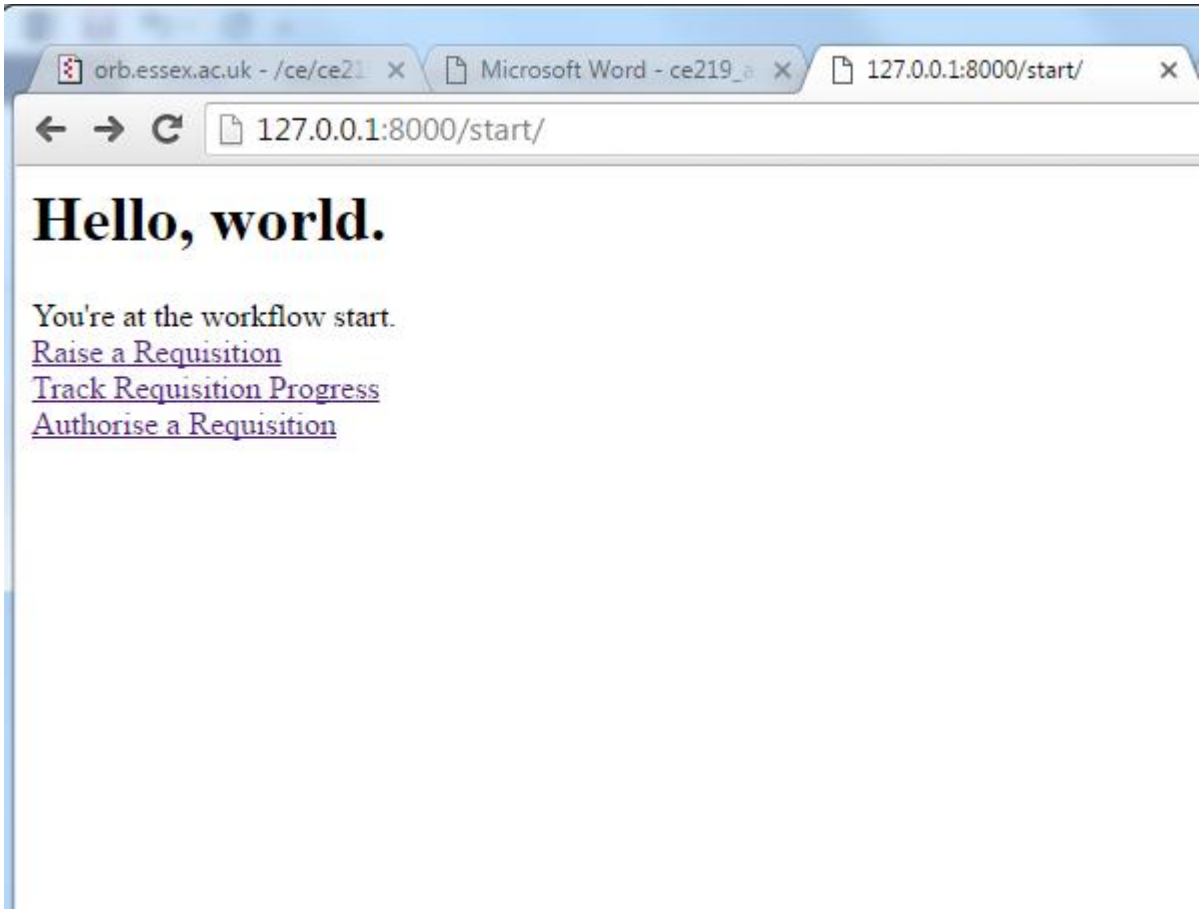
Finally going to requisition_authorization() method it holds a mongodb, gets a auth_token from authorise(), finds it in database and asks user if he is sure that he wants to approve this purchase order. Ordering user to input purchase number again for security reasaons, which then passed to progress() method in which you can see that purchase is authorised.

Mayor problem I encountered is a impossibility to create users of any kind, implementing a login screen, even unavailability to use some of the python features in full file if statements, as Django limits python abilities. Basicly these problems can not be solved because program uses basic http responses which are in pure string format, and using it is useless for any true Django work as non of the Django features can be added to it. Moreover because of these problems system cannot use self numbering system, cannot compare inputs with database components and making software just as a demonstration tool of basic design. With flaws like if there are multiple same number purchase orders they all will be approved at once and putting real time approving time on orders are kind of a struggle as well, because Django is very detailed syntax checking software, so I decided to add only what code should be written like if it actually would allow me to add it nicely.

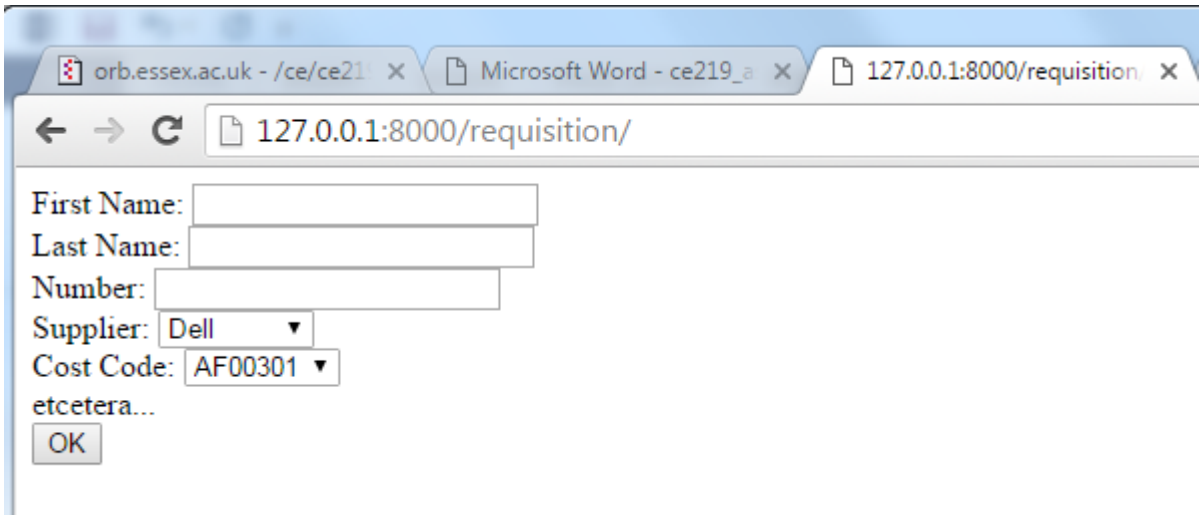
3.2 Another Example

4. Testing and Evaluation

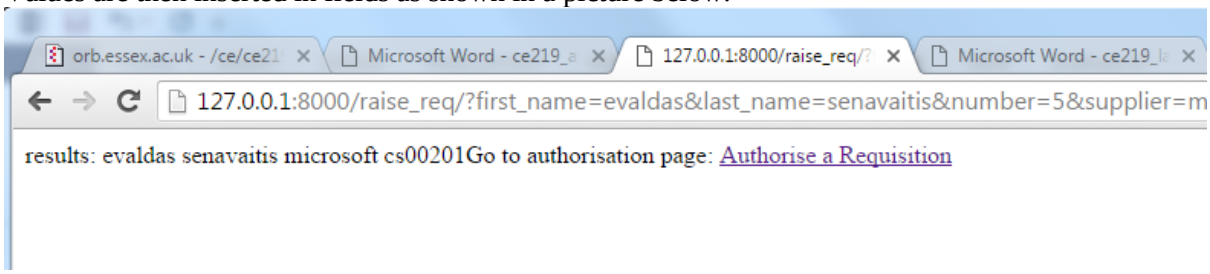
Testing of users is not possible, but I am including screen shots from running solution
Title or Main screen:



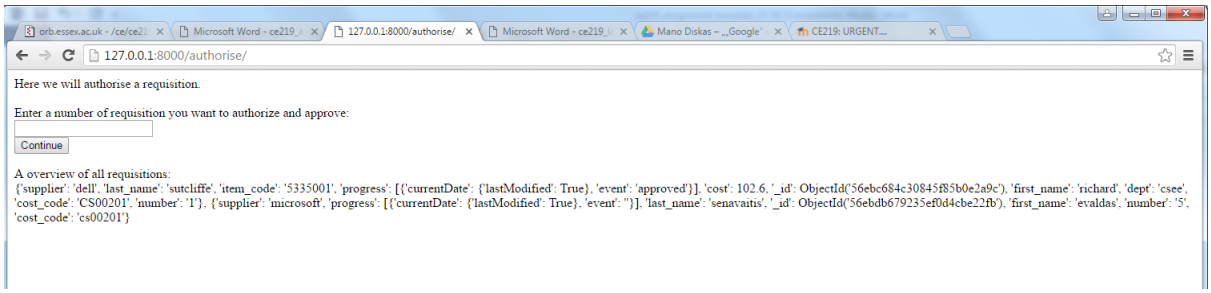
Opens raise a requisition:



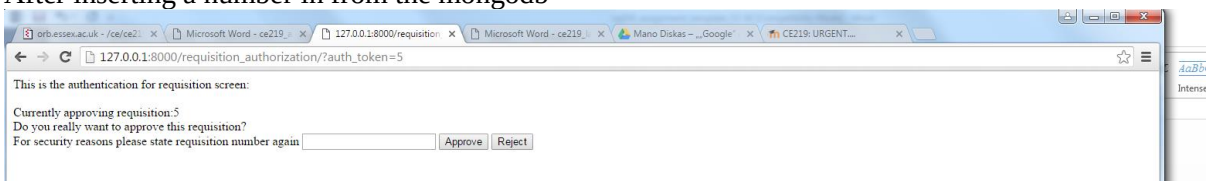
Values are then inserted in fields as shown in a picture below.



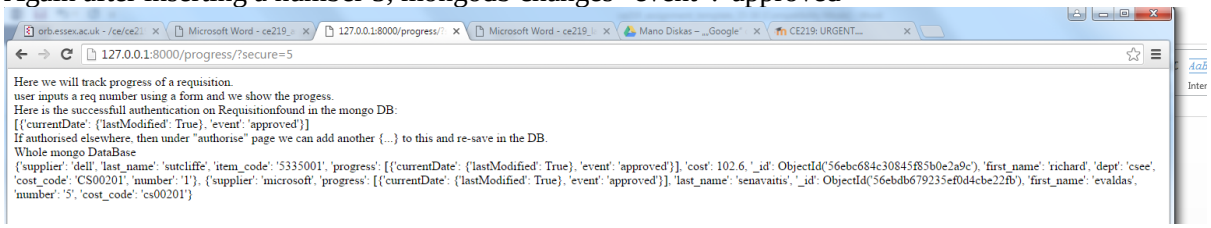
Pressing a authorise a requisition:



After inserting a number in from the mongodb



Again after inserting a number 5, mongodb changes “event”:“approved”



When the report is finished, convert it to pdf (do NOT submit a doc file as this cannot ever be accepted). Use the file name 1234567_my_name_ce219_assigment.pdf.