

Django DBMS Project Report

Software Engineering Team Project



Team members:
Chao Chen, Saheed Adepoju, Yihang Zhao

Fall 2016

Content

1. Project Commitment	3
2. User Stories	4
2.1 Use case card	4
2.2 Activity Diagram	8
2.3 Swimlane Diagram	9
3. Working Code	10
3.1 Data Model Sample	10
3.2 Interface	11
3.3 Code Sample	12
3.4 User Interface Screen	13
4. Project Repository and Tools	14
4.1 Repository	14
4.2 SCRUM Tool	15
5. Weekly SCRUM Sheets	16

1. Project Commitment

The target of Django DBMS project is to develop a web-based platform for small store owner or manager to manage resources and for the staffs to conduct their regular works.

This project is built on top of Django-based web service, therefore it only need to be deployed on specified server instead of to be deployed on different working environment. All the user need are just the internet and web browser.

Below figure shows the project commitment form:

**Prairie View A&M University
Department of Computer Science**

COMP 5423 Software Engineering Project Commitment Form
FALL 2016

This form should be filled by the students and signed by ALL the students. The completed form should be returned to Dr. A. Lodgher by September 21, 2016.

Student Names: Chao Chen Sahed Adepoju Tihang Zhao

Project Title: Django DBMS

Title and Brief Description of the Project:
This project targets on delivering a django-based website for small store owner, manager and staff to manage resources and conducting regular works.

Commitment to the Project by Student:

Student Name	Student Signature	Date	Email	Phone
<u>Chao Chen</u>	<u>Chao Chen</u>	<u>9/18/2016</u>	<u>cohen.rough@gvamu.edu</u>	<u>8329104615</u>
<u>Zhao Tihang</u>	<u>Zhao Tihang</u>	<u>9/20/2016</u>	<u>tong9220@gmail.com</u>	<u>8324741068</u>
<u>Sahed Adepoju</u>	<u>Sahed Adepoju</u>	<u>9/20/2016</u>	<u>sahedade@gmail.com</u>	<u>7134729263</u>

By signing above, we certify that we will do original work for the project and will do the best to complete the above project on time. Our individual failure to work on the project may result in a grade that may prevent us individually and our team members from getting a satisfactory grade for the class. We have indicated our team contact information (Email and phone) and we individually will use this as our primary communication channel, and respond immediately to the communication.

The Team leader for this team is : Chao Chen

2. User Stories

2.1 Use case card

Scenario 1: Account Creation.

Given that user doesn't exist
And can not use the System.
When the user tries to use the Application
The Application allows user to create
Account.(username and Password)
The Application allows the user to log
onto the system.

Scenario 2: Logging out.

Given that a user has logged
onto the System.
When a user clicks log-out
The Application logs out the user.

Scenario 3: Account login.

Given that there is no current user logged in.

Given that there already exist accounts in the system.

Given that the username and password are correctly entered into the Application.

When they are entered

The Application logs them into the system.

Scenario 4: Add New Stock to the Application.

Given that a user is logged in.

Select the workflow menu.

Select the expenditure sub-menu.

Select the stock sub-menu.

Enter the Name, the Price, count and Payment.

Click Submit

The Application will outline that the process was successful.

Scenario 5: Add a new membership Account.

Given that an admin user is already logged in.

Select the workflow - menu.

Select the membership sub-menu.

Select the New-membership Sub-menu.

Enter the Name, Phone number, gender,
Birthday, balance amount, Pin Code, Address.

Click Submit to submit the details.

The application should respond with that
the record has been entered.

Scenario 6: Purchasing an item.

Given that an admin user has logged into the system.

Select the workflow menu.

Select the Purchasing sub-menu.

Select the Purchasing sub-menu.

Select the Stock from the drop-down-menu

Select the Quantity needed.

Enter the Quantity needed.

The application calculates the total amount

due.

Enter the membership ID of the customer.

Enter the Submit button to complete

Select the Transaction.

Scenario 7: @SearchCustomer Query a customer.

Given that an admin user is logged onto the system.

Select the Query Menu.

Select the Query and Edit Sub-menu.

Select the Customer item from the class drop down menu.

Select the attribute you would like to use to search for a customer.

Click on Search

The application displays all users who meet the criteria entered.

Scenario 8: @SearchEdit a customer

Given that an admin user is logged onto the system.

Given that the customer details exist with the system.

Select the Query menu.

Select the Customer item from the class drop-down menu.

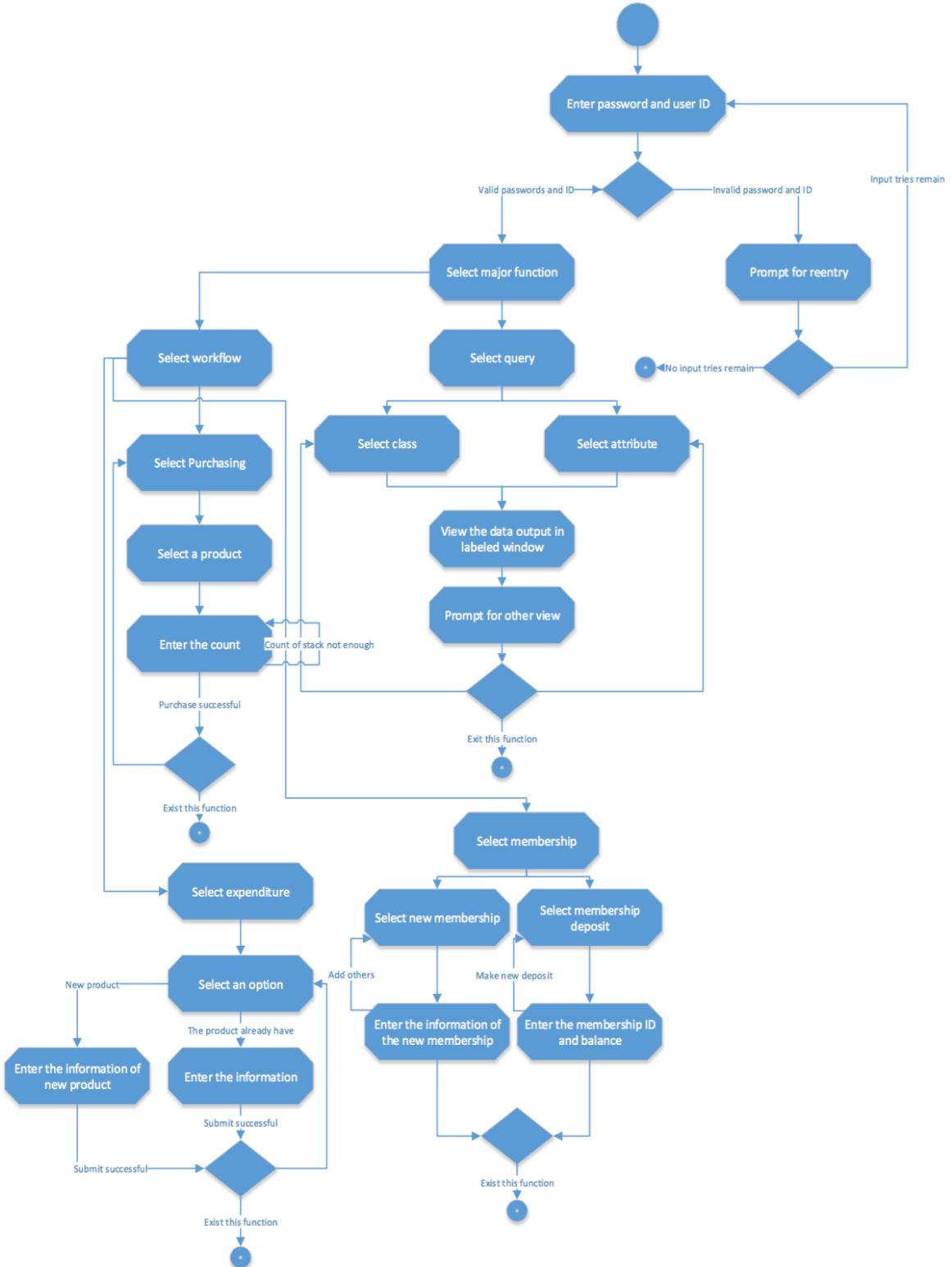
Select the attribute you would like to use to search for a customer.

Search for a customer

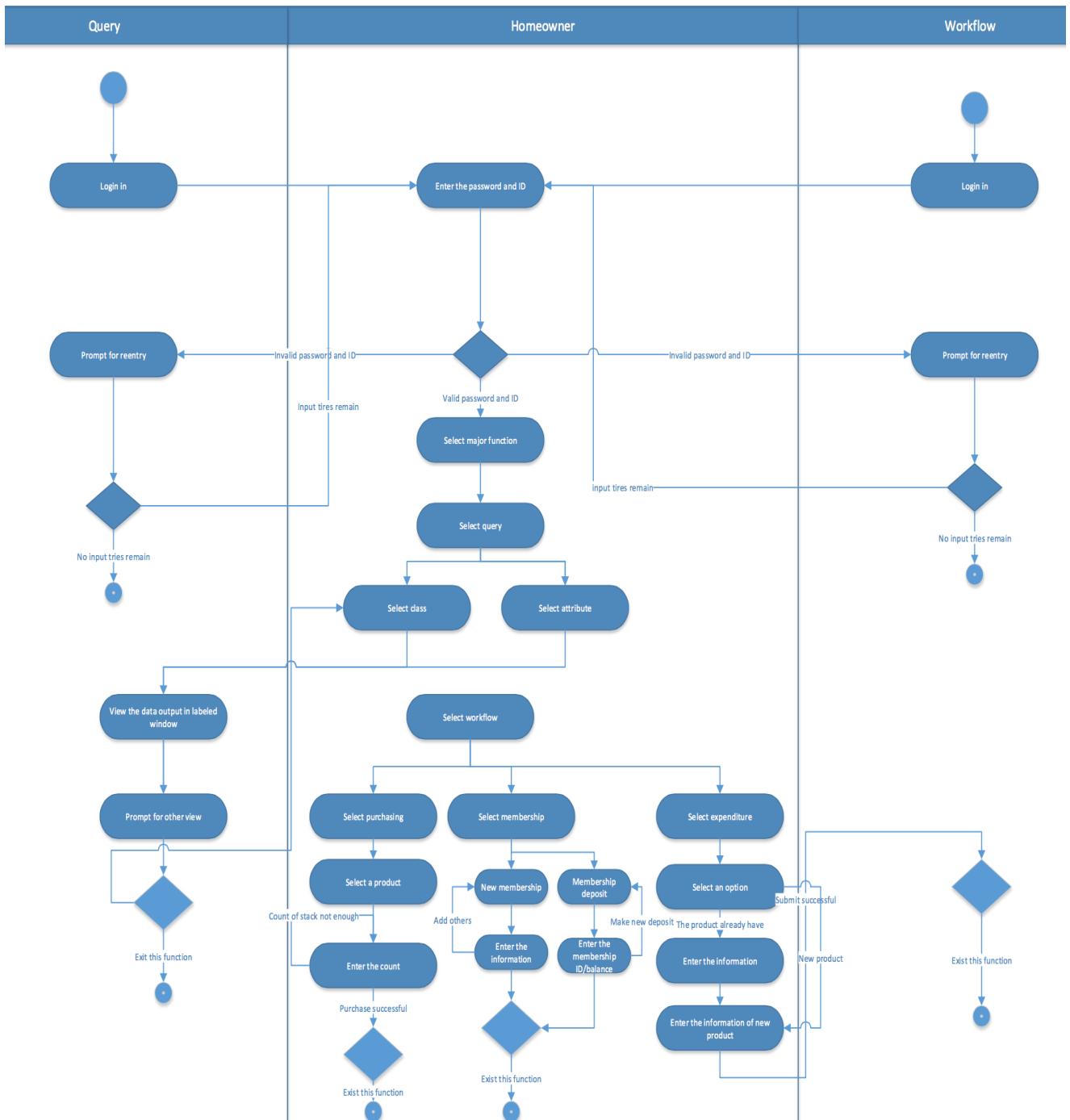
Click on Search.

The application displays all users who meet the criteria entered

2.2 Activity Diagram



2.3 Swinlane Diagram



3. Working Code

3.1 Data Model Sample

```
class Customer(models.Model):
    name = models.CharField(max_length=32)
    phone = models.CharField(max_length=32)
    address = models.CharField(max_length=255)
    gender = models.CharField(max_length=16)
    birthday = models.DateField()

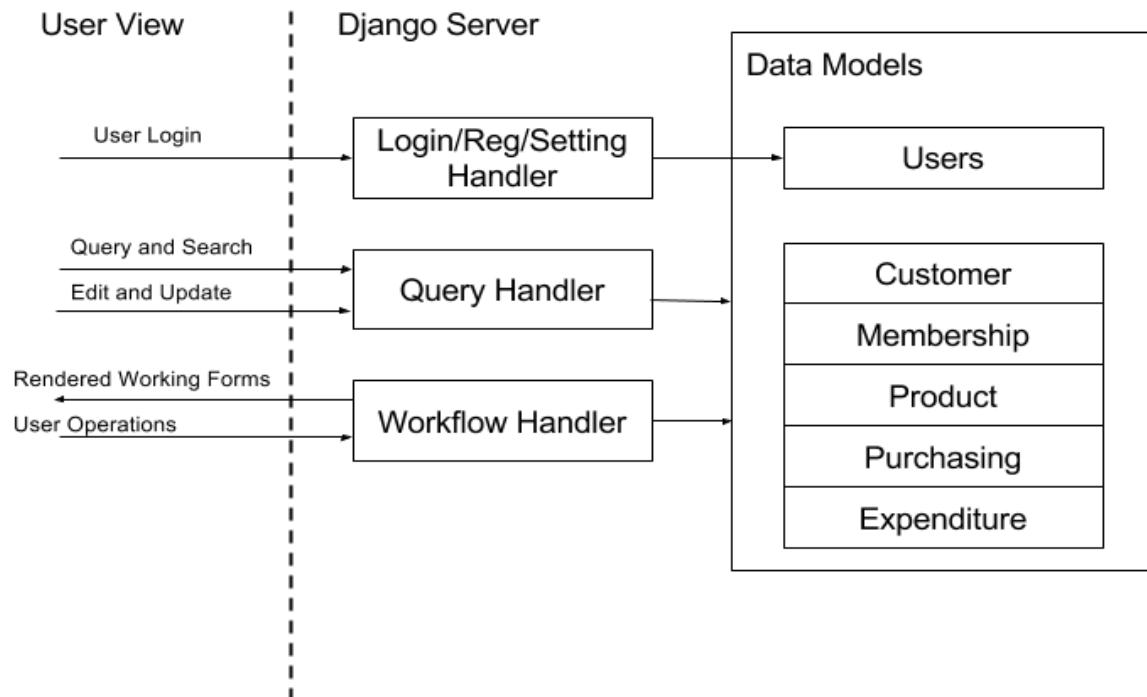
    # cchen @ 2016.09.10:
    # Add unique constraints for name+phone combination
    class Meta:
        unique_together = ('name', 'phone')

class Product(models.Model):
    name = models.CharField(max_length=32)
    count = models.BigIntegerField(null=True)
    price = models.DecimalField(max_digits=19, decimal_places=2)
    details = models.CharField(max_length=255)

class Membership(models.Model):
    regDate = models.DateField(auto_now_add=True)
    balance = models.DecimalField(max_digits=19, decimal_places=2)
    passphrase = models.CharField(max_length=32)
    customerNo = models.ForeignKey(Customer, related_name='membership', null=True, on_delete=models.SET_NULL)
```

3.2 Interface

Below figure shows the overview of the web application's framework, it also demonstrate how the web view handler interact with the user view an data models.



3.3 Code Sample

Below code section demonstrate how to perform user operations on data model, in this case, it's creating or modifying product entry:

```
# Products new/update handler
@login_required
def exp_product_reg(request):
    resp = 'OK'
    needUpdate = True

    productNo = request.POST.get('product')
    count = request.POST.get('count')
    details = request.POST.get('details')
    print productNo, count, details
    try:
        if productNo == "new":
            new_name = request.POST.get('product_new')
            new_price= request.POST.get('product_price')
            prod = Product(name=new_name, price=new_price, count=count)
            prod.save()
            productNo = prod.id
            needUpdate = False

            payment = request.POST.get('payment')
            prod = Product.objects.get(id=productNo)
            exp = Expenditure(payment=payment, details=details, productNo=prod, count=count)
            exp.save()

            if needUpdate:
                prod.count = prod.count + long(count)
                prod.save()

    except:
        print >> sys.stderr, traceback.format_exc()
        resp = 'exp product reg error'

    return HttpResponse(resp)
```

3.4 User Interface Screen

This project finally delivered a website that could be deployed on web server or local computer. Below screenshots show some user interface of this application:

The screenshot shows the Waka application's Expenditure module. The top navigation bar includes links for Home, Query, Workflow, About, and a user account for 'cchen'. On the left, there are three categories: Purchasing, Membership, and Expenditure, with Expenditure being the active tab. The main area contains a 'Stock In' form with fields for 'Stock' (a dropdown menu with 'New stock'), 'Count' (a text input with value '30'), and a 'Payment' section containing a list with items '20' (highlighted in yellow), '2' (highlighted in blue), and '20' (highlighted in light blue). A 'Details' link is visible next to the payment items. A 'Submit' button is at the bottom.

The screenshot shows the Waka application's Query and Edit module. The top navigation bar includes links for Home, Query, Workflow, About, and a user account for 'cchen'. On the left, there are two tabs: 'Query and Edit' (active) and 'Backup and Restore'. The main area features a search interface with 'Select Class:' set to 'Customer', 'Search attribute:' set to 'name', 'Attribute value:' set to 'Chao', and a 'Search' button. Below the search is a table with columns: * (checkbox), id, name, phone, address, gender, and birthday. One row is shown with values: edit, 1, Chao Chen, 8326431256, 5625 Bissonnet ST Apt 207, Male, 2016-09-01.

The screenshot shows the Waka application's About page. The top navigation bar includes links for Home, Query, Workflow, About, and a 'Login' link. The main content area features a large 'About Waka' heading and a 'Waka Manager' section with a 'Learn more >' button. A modal window is open in the center, containing tabs for 'Login' and 'Sign Up'. The 'Sign Up' tab is active, showing fields for 'Username' (with placeholder 'Username'), 'Email address' (with placeholder 'Email'), 'Password' (with placeholder 'Password'), 'Confirm Password' (with placeholder 'Password'), and a 'Register' button. A close button is located in the top right corner of the modal.

4. Project Repository and Tools

4.1 Repository

This project uses GitHub as the source code control tool. Below figure shows the screenshot of the source code repository:

The screenshot shows the GitHub repository page for the project 'cchen1983/DjangoDBMS'. The repository has 3 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was made 6 hours ago. The repository page includes tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Pulse, Graphs, and Settings. It also features buttons for Unwatch, Star (0), Fork (0), and Clone or download.

File	Description	Time Ago
contacts	Add first stage development source.	13 days ago
home	Add first stage development source.	13 days ago
login	Update code comments	6 hours ago
query	Update code comments	6 hours ago
static	Update code comments	6 hours ago
templates	Add first stage development source.	13 days ago
waka	Add first stage development source.	13 days ago
workflow	Update code comments	6 hours ago
.gitignore	Add first stage development source.	13 days ago
ReadMe.txt	Initialize the django project.	13 days ago
get-pip.py	Add first stage development source.	13 days ago

4.2 SCRUM Tool

For project management purpose, we use Scrumwise, an online SCRUM platform , to manage all the tasks and resources by evenly distributed them to multiple sprints. Below screenshot shows the situation in one of the sprints:

The screenshot displays two views of the Scrumwise interface. The top view shows a 'Sprint Board' with two sprints: Sprint 2 and Sprint 3. Sprint 2 is completed with 9 days completed. Sprint 3 is in planning with 3 days left. The bottom view shows a detailed 'Backlog item' for 'Database Design and Implementation'. The backlog item has a name, description, rough estimate (60 points), type (Feature), and status (In progress). It is assigned to Chao and is part of Sprint 1. A progress bar indicates 100% completion.

Sprint	Task	Team	Duration	Status	Progress (%)
Sprint 2	Workflow page design.	DJTeam	2 d	Sprint completed	100%
	Query Page Design	DJTeam	2 d	Sprint completed	100%
Sprint 3	Implement Query page functionality	DJTeam	3 d	Sprint completed	100%
	Django interfaces for Data Models	DJTeam	2 d	Sprint completed	100%
	Implement Workflow page functionality	DJTeam	4 d	Sprint completed	100%

Backlog Item	Name	Description	Rough Estimate	Type	Status	Assigned To	Progress (%)
Web server deployment	Database Design and Implementation	1. Design and implement the database 2. Testing and debug	60 points	Feature	In progress	Chao	100%
Database Design and Implementation			2 days		Done		
Workflow page design.			2 days		In progress		0%
Query Page Design			2 days		In progress		0%

5. Weekly SCRUM Sheets

COMP 5423 Software Engineering Processes, CS Dept, PVAMU
Fall 2016 SCRUM MASTER DAILY/WEEKLY FORM

This form must list the date of each week, tasks assigned to each team member for the week, and the answers to the three questions that the Scrum Master must ask each team member every day. All members MUST sign the form. Entries in this form WILL contribute towards individual member grade changes. Incorrect information WILL lead to grade degradation of entire team. This form MUST be submitted to the instructor EVERY week by the SM to the instructor at beginning of class. Each team will project this form in class, to present their progress in class each week.

Project Name: Django DBMS

Date of each week:	9/8/16	9/9/16	9/10/16	9/11/16	9/12/16	9/13/16	9/14/16	Time set for call/ meeting each day: 10:00 AM PM	
Master Name:	<u>Chao Chen</u>							SM: All tasks accomplished?: Yes No	SM Sign: <u>Chao Chen</u>
Tasks Assigned for the week:	Design Database models; Server Scripts for Login, query, workflow								
What did you do?	Database Modeling	Database Modeling		Design the Login App.	Design Workflow App.	Query App.	Test and debug		
What obstacles you had?				Ajax error troubleshooting			Serve ajax request error		
What do you plan to achieve?	Design Database	Implement Database		Finish the login function	Finish the Workflow view script	Finish the scripts..	debug		
Date of each week:	1/16	1/16	1/16	1/16	1/16	1/16	1/16		
Member 2 Name: <u>Sachet Addepudi</u>	SM: All tasks accomplished?: Yes No							Mem 2 Sign: <u>Sachet Addepudi</u>	
Tasks Assigned for the week:	Implement Query App UI.								
What did you do?	Learning and practice Query UI.	Design Query UI.	Implement Query UI.		Implement Query UI.	Debug and Testing			
What obstacles you had?			Communication issue with Django		Testing on Django not pass	Integrate with Django Server			
What do you plan to achieve?	Know how to use Query	Finishing UI Prototype	Fix ajax request fail	Fix Data query issue.	Fix Django error				
Date of each week:	1/16	1/16	1/16	1/16	1/16	1/16	1/16		
Member 3 Name: <u>Yihang Zhao</u>	SM: All tasks accomplished?: Yes No							Mem 3 Sign: <u>Zhao Yihang</u>	
Tasks Assigned for the week:	Implement Workflow App UI.								
What did you do?	Design Workflow UI.	Implement UI.	Learn practice Query API.		Implement Purchasing Function		Debug and test		
What obstacles you had?			Some jQuery API not work. max api				The result of data import not correct		
What do you plan to achieve?	Coding	Fix Query usage issue.	Know how to fix problem	Test and troubleshoot			Fix database operation error		

COMP 5423 Software Engineering Processes, CS Dept, PVAMU

Fall 2016 SCRUM MASTER DAILY/WEEKLY FORM

This form must list the date of each week, tasks assigned to each team member for the week, and the answers to the three questions that the Scrum Master must ask each team member every day. All members MUST sign the form. Entries in this form WILL contribute towards individual member grade changes. Incorrect information WILL lead to grade degradation of entire team. This form MUST be submitted to the instructor EVERY week by the SM to the instructor at beginning of class. Each team will project this form in class, to present their progress in class each week.

Project Name: Django DBMS

Time set for call / meeting each day: 10:00 AM PM

Date of each week:	9/15/16	9/16/16	9/17/16	9/18/16	9/19/16	9/20/16	9/21/16
Master Name:	Chao Chen						
Tasks Assigned for the week:	Integrated test and debug Documentation.						
What did you do?	Test / Debug	Test / Debug	Test / Debug		Documentation	Documentation	
What obstacles you had?	The tuples rendering not correct	IE compatibility issue.					
What do you plan to achieve?	Fix tuples render issue.	Fix unique binding for customer entry	Fix browser compatibility issue.		Finish the docs.	Finish the docs.	

Date of each week:	1/16	1/16	1/16	1/16	1/16	1/16	1/16
Member 2 Name:	Yihang Zhao						
SM: All tasks accomplished?: Yes No	Yes						
Tasks Assigned for the week:	Workflow page integrated test and debug Documentation.						
What did you do?	Test / Debug	Test / Debug		draw activity diagram	draw swimlane diagram	redraw swimlane diagram	
What obstacles you had?	Workflow add new protocol	submit part bug		figure out team members work.		principle here some problem	
What do you plan to achieve?	Fix it	using online example, and change the major function		ask them about function they did	Finish it	redraw it according the textbook.	

Date of each week:	1/15/16	1/16/16	1/17/16	1/18/16	1/19/16	1/20/16	1/21/16
Member 3 Name:	Saleem Adepoju						
SM: All tasks accomplished?: Yes No	Yes						
Tasks Assigned for the week:	Query page integrated test and debug Documentation.						
What did you do?	Test / Debug	Test / Debug		Documentation	Develop user stories.	Developed user stories.	
What obstacles you had?	Query overwriting what was planned	Query wasn't working for static calculator					
What do you plan to achieve?	Fix the query b1 customized media.	Fix the engine for the calculator		Finish the documentation	Complete the user stories.	To work on test -> final integration to new structure.	