

EPICS Final Report

Team: Butler Theatre

Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal
Fall 2019 Semester

Table of Contents

Summary/Abstract	3
Chapter 1: Introduction	4
Chapter 2: Requirements Specifications	6
Chapter 3: System Architecture	7
Chapter 4: Design	9
Chapter 5: Implementation	11
Chapter 6: Quality Assurance and Testing	12
Chapter 7: Project Organization and Management	13
Chapter 8: Future Work	21
References	22
Appendices	23

Abstract

This final report encompasses all of the work that the Butler Theatre group did during the fall 2019 semester. We worked on building a project for the Butler University Theatre department in order to help create a program that would allow them to organize and keep track of what is currently in their warehouse. Currently, they have over 15,000 costume pieces held in a warehouse under the Butler University parking garage. While their items are sorted and organized, there are no current records of all of the items themselves. This is a hinderance for the theatre department, as it is constantly a guessing game as to what they do or do not have.

Our project revolves around finding a solution to this problem and getting starting at creating this solution. As time went on, we discovered that this project is quite complex and will take more than one semester to complete. Therefore our goal was to set up a strong foundation that others can build off of in the future.

This report consists of eight main parts:

1. Introduction
2. Requirements Specifications
3. System Architecture
4. Design
5. Implementation
6. Quality Assurance and Testing
7. Project Organization and Management
8. Future Work

Along with this, there is a reference/bibliography section and an appendix at the end.

Our team took this semester as a large learning experience as we were all new to the programs that we decided to use, it was a tough but beneficial class that we all enjoyed taking.

Chapter 1 – Introduction

Our team for EPICS is called Butler Theatre and is comprised of five people: Eromo Aligbe, Maya Grandstaff, Kameron Leisure, Geralyn Miller, and Sam Royal. The team is from a set of completely different backgrounds which allowed us to bring unique skills to the table in order to make this project successful.

Developer Background Overview:

- Eromo Aligbe, Technical Lead – sophomore Computer Science major
- Maya Grandstaff, Team Lead – senior Marketing and Management Information Systems majors
- Kameron Leisure, Database Technician – junior Computer Science major and Data Science minor
- Geralyn Miller, Documentation and Website Lead – senior Actuary Science major and Data Science minor
- Sam Royal, Client Liaison – senior Management Information Systems major

The client that we worked for was the Butler Theatre department, which is considered to be one of the best theatre programs in the nation and is highly respected by many. Our main point of contact was Megan Wiegand, who manages all of the costumes for the Butler Theatre department.

Problem:

The Butler Theatre department has a large collection of clothing that they use for the shows that they perform, but there is no organizational system set in place that people can use to gather information about which items they have in their warehouse.

Objectives:

- Create a system that would allow for both storing and accessing information about the garments in the costume warehouse
- Build a strong foundation for this project so that other groups can work off of this in future semesters

One of the objective that we have set is to build a strong foundation because our group is starting this project from scratch, and it is crucial to start strong and solid and allow ourselves and others build upon this.

Our motivation for this project was founded upon seeing how good the Butler Theatre department can put on shows, and wanting to help them optimize their practices so that they can prepare for different shows much more efficiently. Since they have no clear record keeping system put in place currently, it can be difficult to pull various items for shows and see what they do and do not have, which is why we wanted to help them out.

Again, this project was brand new and was not carried on from a previous semester, so our approach was to set up a foundation that teams can then build upon in future semesters. In

order to do so, we broke into two different sub-teams to conquer the two sides that were needed for this project: a back-end database and front-end application. This will be explained further in depth later on in this report.

The organization of this report will consist of eight chapters, including this one, that will cover an Introduction, Requirements Specifications, System Architecture, Design, Implementation, Quality Assurance and Testing, Project Organization and Management, and Future Work. Within each section, the overall topic will be broken down into further detail as it relates to our project. The table of contents at the beginning of this report gives an overview of where each section starts.

Chapter 2 – Requirements Specification

This section will cover the requirements specifications that this project will need, broken down into functional and non-functional requirements. Along with this, the assumptions and constraints of this project will be discussed.

Functional Requirements:

- The project will need to have two main parts: a front-end application that the user will interact with and a back-end database that will store all of the information
- Connection between the two ends that is able to run queries between
- Rest API to facilitate communication
- Different authorization levels (ex: able to pull information vs. able to add or edit clothing entries)
- Authentication login process so that only authorized users can access the information
- Reporting requirements: a proper format to give information that they want to users

Non-Functional Requirements:

- Quick response time for queries and information pulling
- Scalability - Able to handle thousands of clothing entries, the Butler University Theatre Department has approximately 15,000 pieces in their warehouse
- Reliability – will not crash when being used
- Data Integrity – secure data that cannot be accessed by everyone, only by authorized users
- Usability – easy to use by anyone, client specified that they are not tech-savvy and want the program to be as hands-off as possible
- Manageability – the program should be easy to maintain or add/edit new clothing entries by the users

Assumptions:

- This program will need to be extremely user friendly as the client wanted to be able to use it easily without having any experience with coding
- Since this is the beginning of the project, this semester will focus on building a foundation that others can build off of, and the program is only starting and will need to be worked on in semesters going forward

Constraints:

- Feasibility of what work can be accomplished in one semester
- Programming languages – the whole team is relatively new to the languages that are used this semester and it will be a learning process, might slow down the velocity that this project can be completed at

Chapter 3 – System Architecture

This chapter will cover the system architecture, including its services, components, relationships, interaction between components, and responsibilities of each component. It will also cover the architectural decomposition and style, and the system platforms.

Services:

The services for this project are similar to the requirements that are needed, and can also be considered the features. The main feature of this project is that it is user-friendly by having two main components: a front-end application and a back-end database. This was because of the client's request to have an application that is quite hands-off so that they do not have to directly interact with the database. Along with this, another feature that we have is that the database is hosted directly on Butler's server so that it can be accessed directly by any device on their WiFi using an IP address so that changes can be made by any user.

Components and their Responsibilities/Functionalities:

There are two main components to this project: a front-end application and a back-end database. By using Django for the front-end application we are also able to create a rest API simultaneously, reducing the need for a third implementation language and separate third part of the project to create. The front-end application is what the users will interact with rather than having to navigate a foreign and complex database.

Along with this, there is a back-end database that stores all of the information that can then be pulled to the application through running queries. This database will communicate with the front-end application through the rest API, as shown in the use-case in the section 'Relationship and Interaction between Components'.

Architectural Decomposition and Style:

From our research, it seems that the style of architecture used for this project was client-server. This is because the clients would be able to access the front-end of the application from their devices, but then the database is hosted on the Butler servers rather than on their devices as well.

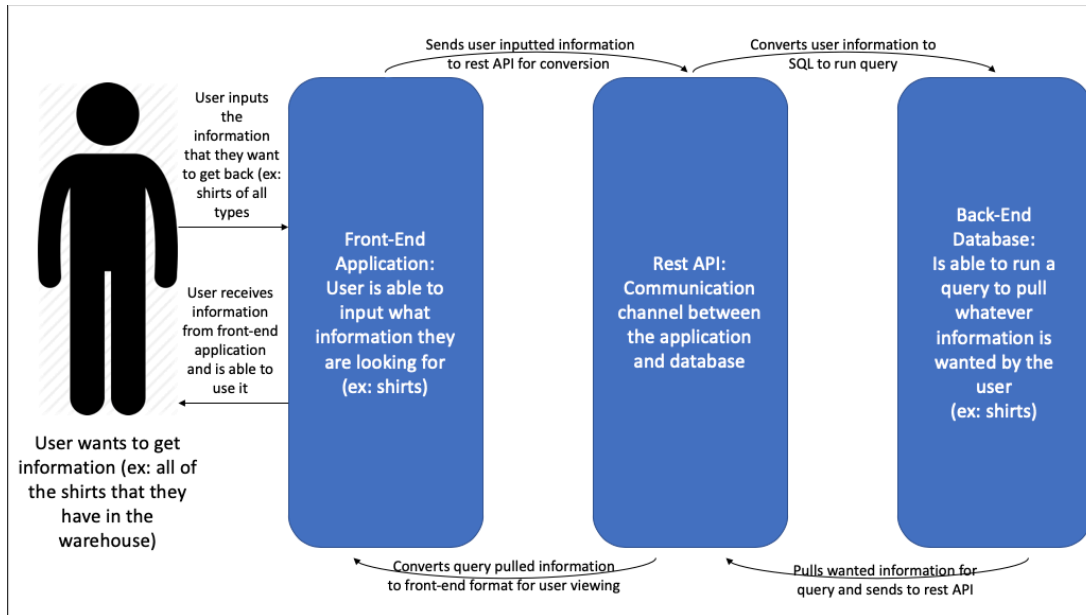
System Platforms:

The hardware used for this was quite simple: computers. We primarily worked on Windows for an operating system as we had complications early on using iOS. By switching over to working solely on Windows the process was much easier and optimized, and thus we stuck with this until the end of the semester.

As for software, the back-end database team used MySQL through myPHPadmin, which could be accessed anywhere online on-campus as it was hosted through a Butler server. This made it easy for the team and easy for anyone who wants to work on the project going forward.

Relationship and Interaction between Components:

The best way to represent the relationship and interaction between components is by using a use-case diagram. The use case below has the user wanted to gain information using the system in order to get information of all of the shirts that are in the warehouse.

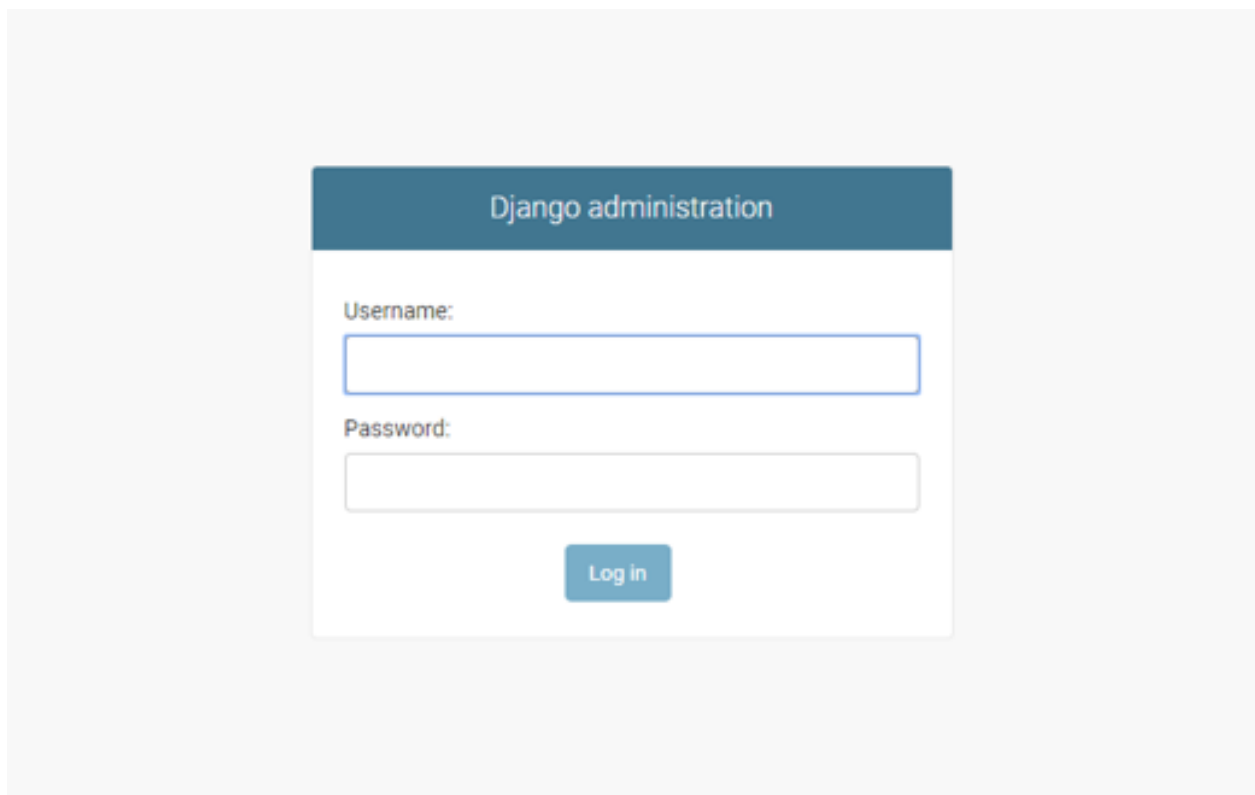


Chapter 4 – Design

The design chapter will be one of the shorter chapters in this report, as it entails only a few pages that we created. Since this semester our team focused on setting the foundation for this project, there is a lot of room for teams in the future to put their own design stamp on how they want everything to look.

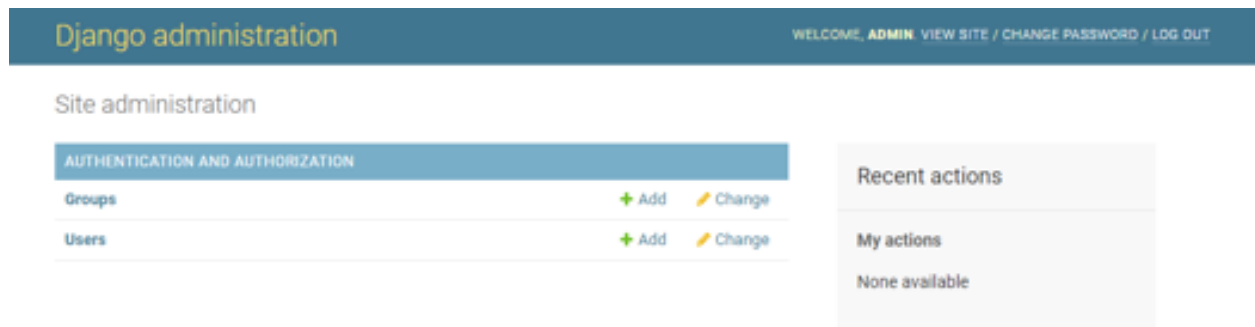
The main user interface that our team was able to create is the administrative page for the Django front-end application. This is where users and groups of people can be managed, and in the future tables of information can be added about different types of garments. The design and layout of both of these pages is quite simple so that they are easy to manage and are not overwhelming to look at.

Screengrabs for front-end application:



Description:

The login page is an authorization measure put in place so that only certain people can access the actual administrative page. There is a general admin user, our client Megan, and our team members that have login credentials to get past this.

**Description:**

The administrative page allows people to manage two main components: the authentication and the information tables. For authentication, it allows users to add, remove, and edit the privileges of different users. This supports some of the features that we were looking for within the program.

Along with this, there are information tables that can be added in. There are currently none on the site that we made but they are easy to add in by others that will work on the project in the future. These tables will mimic the ones that are in the database so that entries can be added in or changed.

While some user interface design has been inputted, again there is a lot more room for other groups to build off of this with their design influence. We do recommend keeping the layout aesthetic as simple and as clean as possible, so that it will not overcomplicate what the site looks like and also allows for ease of use by the users.

Chapter 5 – Implementation

There are two main components to implementation: the selection of implementation languages and organization of the code base. Coding standards/comments used and implementation process/distribution of work are omitted because the coding standards are quite basic and comments were not very necessary for the work that we completed, along with the distribution of work and velocity of the team being discussed in Chapter 7.

Selection of Implementation Languages:

There are two implementation languages that we used this semester to set up the program: Django and MySQL. This is because there were two components: the front-end application and the back-end database.

For the front-end application Django was selected at the language of choice. There was a debate between this and Wordpress in the beginning, however we decided to use Django in the end. This is because it was more user friendly to work with to code and allowed for more structure compared to Wordpress. Structure was important for the team as we were new to these languages and did not want a completely freeform program to begin with.

For the back-end database, MySQL was used. In the beginning SQLite was used however, it was important to switch over to My SQL so that there was more room for item entries, since the Butler Theatre department has upwards of 15,000 garments in their warehouse and we did not want the system to crash. This switch was easy to make since it was done early on in the semester.

Organization of the Code Base:

In the beginning of creating the front-end application code, it was all done on Geralyn's computer. However, by the end of the semester we managed to migrate it to Thomas.butler.edu, and uploaded all of the code written to our team's GitHub so that future teams could access this and create their own applications as needed.

The MySQL is currently on a server hosted on Butler University's server and can be accessed by anywhere on the Butler University campus. This means that anyone now or in the future can view and edit on the myPHPadmin site that is attached to the MySQL server. All of the information that is needed in order to access this as well is on the team's GitHub page. The link to the GitHub page is also listed on the Butler Theatre EPICS site for easy access for anyone that might need it in the future.

The main takeaway for the code base is that leaving this semester, it is housed primarily in the Butler-Theatre GitHub page, which is accessible via our project archives.

Chapter 6 – Quality Assurance and Testing

This chapter will be the shortest by far, as there are currently no quality assurance and testing measures put in place. Again, our goal was to create a strong foundation so that other groups in the future can build off of this and bring the project to fruition. Because of this, our group did not get to the stage of quality assurance and testing of the program.

However, in place of this, we have provided examples of quality assurance and testing that should be completed in the future:

- See if there are any duplicated ID numbers for garments
- Test inputting each type of garment, editing, and deleting garments, then checking to see if these changes are reflected properly in the database
- Quality assurance: which information fields should be required or optional
- Error checking to see if/what kind of message will pop up in an instance that something is filled out incorrectly
- What the output will look like for the queries that are ran

While the list that is given is not all-encompassing of all quality assurance and testing that must be done for a successful program, it does give an overview of what these procedures should look like when the time does come for this step.

Chapter 7 – Project Organization and Management

Team Organizational Structure and Work Breakdown

The team's organizational structure was quite relaxed and everyone was empowered to do their own thing during class time. The group leader, Maya, would do check-ins and see if anyone needed help and made sure that tasks were being done.

From here, the group broke down into two primary groups: working on the front-end and working on the database. Eromo Aligbe and Kameron Leisure worked on the back-end database development, and Maya Grandstaff, Geralyn Miller, and Sam Royal worked on the front-end application development. Each team checked in with each other to ensure that progress was being made and to see if there was any help needed elsewhere, and this breakdown seemed to work the best for the team's functionality.

Contribution by Team Member:

In their own words, each team member explained what their contribution was to this project to show their growth throughout the semester. Detailed below are each of their experiences:

Eromo Aligbe, Technical Lead:

What I did this semester: learned the basics of python to understand Django in case the front-end needed help. Learned mysql in order to create tables within the myphpadmin server and strengthened my understanding of the command line. Oversaw the creation of the server set up using ubuntu, which improved my overall understanding of servers and how they connect and host databases.

Maya Grandstaff, Team Leader:

As the Team Leader, I was responsible for making sure that everything was running smoothly for the team, keeping open lines of communication, and helping out wherever needed. I have lead groups in the past but this was a different challenge as it was creating an entirely different project from scratch. Overall I tried to lead by empowering everyone else on the team to do what they deemed to be the most important on that day but still keeping them on the right path along the way. We did stand-up meetings at the beginning of each class to check in with each other and see if anyone needed more help on each day. Along with this, I worked heavily on the front end with Geralyn, setting up the Django project and administrative page. This too was a next experience as I have never worked in Django before and this turned out to be difficult than I initially thought. However as time went on and through a lot of trial and error we started to make progress and I am happy with what our group accomplished this semester.

Kameron Leisure, Database Technician:

My job as database technician was to maintain the development of, and see to the creation of the database needed by our project. Before creating any databases, my first task was to develop a schema that the database would pilot off of. This took all of Sprint 1 and most of Sprint 2, as I had feared that the addition of an "Others" table to the database would prove more destructive than beneficial in the long run. After calming that fear, I then moved onto

installing SQLite and creating a database there. This went surprisingly smooth, however the group elected to have a SQL server set up so that anyone may see/manipulate data whenever they wish. So I, alongside the assistance of Nathan Partenheimer, got a SQL server running and established an admin account so that our database is safe to use without outside interference. Overall this project proved to be a little more than I could chew. With having nothing set up before hand, I felt like we were doing very little when we were actually working on the most important step: establishing a starting point for every other group to take off from. The database we have set up here will go on to become something great, but it could not happen until we took that first step, which I am proud of nonetheless.

Geralyn Miller, Website and Documentation Lead:

This semester I worked with Django, and really had to do a lot of learning outside of class since I didn't previously know that language. This class helped me learn to teach myself and develop new ways of learning independently. All of us did a lot of researching at the beginning of the semester to determine which programs we'd use for this project. The front end was originally housed on my computer (before we eventually got it onto thomas), so I would get everything set up before each class, in order to maximize our class time. I worked on the website and kept the documentation up to date. I also worked with Maya on the front end, and we edited/wrote code in order to customize it for the theatre department and their specific needs.

Sam Royal, Client Liaison:

As client liaison, I kept in touch with our Client, Megan Wiegand from the Butler University Theatre. To do this, I would contact her once a week and let her know of an update on where the team was at, and overall I was the contact point between Megan and the team in regards to any communication or meeting scheduling. Along with this, I offered support to the front-end development of our application and the planning of the semester along with planning for where the project will go in the future.

Project Management Process:

Towards the beginning of the semester the team had trouble starting the project, however as time went on the velocity of the team greatly increased. Much of this was because the team was new to the languages that were used and it was a struggle to begin working in them. However, when we made the initial jump from planning and learning to creating, the velocity of the team greatly increased.

Along with this, the team over time went from having lofty, ambitions goals to more feasible steps to be made along the way. This helped us see progress much more easily and helped with our overall project management. Sprint goals were also used to measure progress made throughout the semester.

Weekly Status Reports (WSRs):

At the end of the week each semester, a Weekly Status Report (WSR) was written to reflect on the progress made. There were three main components used in order to create these: work completed, road blocks, and to be completed. This allowed us to track what was

done throughout the semester and take a look at what still needs to be worked on. Each of the 11 WSRs completed are attached below:

WSR 1:**Weekly Status Report**

Project: Butler Theater

Team Members: Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal

Week Of: September 23 - 27

Completed:

- Client site visit and meeting to establish project
- Meet with IT to go over framework and compatibility with Butler for future use
- Started database schema

Road Blocks:

- Deciding which framework to use

To Do:

- Establish project guidelines and tentative deadlines
- Decide on a framework
- Begin to build project

WSR 2:**Weekly Status Report**

Project: Butler Theater

Team Members: Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal

Week Of: September 30 – October 4

Completed:

- Established which programs are going to be used for the project
 - SQLite and Django
- Began research of programs

Road Blocks:

- Getting familiar with programs before making progress on actual project work
 - Tackling this by doing research over various programs

To Do:

- Begin work on project in Django and SQLite
- Create presentation for Sprint 1 – due next Wednesday

WSR 3:**Weekly Status Report****Project:** Butler Theatre**Team Members:** Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal**Week Of:** October 7 – 11

Completed:

- Sprint 1 – goals and presentation for class
- Began learning about Django and Python

Road Blocks:

- Need to learn more about Django and Python before working on project-specific tasks

To Do:

- Begin working on Sprint 2 goals
- Finish database schema
- Start creating prototype of restful API

WSR 4:**Weekly Status Report****Project:** Butler Theater**Team Members:** Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal**Week Of:** October 14 - 16

Completed:

- Established sprint 2 goals
- Follow-up client meeting to get further details for database

Road Blocks:

- Setting up Python and Django for starting rest API and connecting to database
- Need to complete database schema

To Do:

- Complete database schema
- Set up Python and Django to begin front-end work
- Have a safe and fun fall break!

WSR 5:**Weekly Status Report****Project:** Butler Theater**Team Members:** Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal**Week Of:** October 21 - 25

Completed:

- Set up Python and Django to start building rest API
- Team evaluations

Road Blocks:

- Need to connect database to rest API
- Need to complete database schema in order to begin building database

To Do:

- Connect database to rest API
- Start building database
- Review evaluations as a team in order to ensure future success
- Start building rest API

WSR 6:**Weekly Status Report****Project:** Butler Theater**Team Members:** Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal**Week Of:** October 28 – November 1

Completed:

- Reviewed peer evaluations
- Worked with Nate on how to build rest API using Django

Road Blocks:

- Possibly switching directions for front-end
 - The team is current deciding between whether or not to use Django or Wordpress for the front-end of the system

To Do:

- Decide which program to use for the front-end
- Keep building rest API
- Building database

WSR 7:

Weekly Status Report**Project:** Butler Theater**Team Members:** Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal**Week Of:** November 4 - 8

Completed:

- Started building database
- Updated client on progress so far
- Continuing to build rest API

Road Blocks:

- Still a possibility of changing paths to use Wordpress instead of Django

To Do:

- Set up MySQL server with Nate from IT
- Work on getting front end to the point of being able to run queries

WSR 8:

Weekly Status Report**Project:** Butler Theatre**Team Members:** Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal**Week Of:** November 11 – 15

Completed:

- Chose an official path for front-end application – using Django instead of WordPress
- Set up database on Butler server

Road Blocks:

- Need to get front-end application off of locally hosted computer to a place that can be accessed by more than one device in order to move forward

To Do:

- Connect database to front-end of application
- Get front-end application off of locally hosted device

WSR 9:**Weekly Status Report****Project:** Butler Theater**Team Members:** Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal**Week Of:** November 18 – 22, 2019

Completed:

- Met with client in order to go ask further questions and give Butler University Marketing department a tour and interview
- Began working on final poster

Road Blocks:

- Database and front-end application are not yet connected
- Get front-end application off of locally hosted device

To Do:

- Get front-end application off of locally hosted device
- Connect database and front-end application

WSR 10:**Weekly Status Report****Project:** Butler Theater**Team Members:** Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal**Week Of:** December 2 - 6

Completed:

- Got front-end application off of locally hosted device and onto Thomas.Butler.edu
- Began working on connecting front-end to database
- Created Sprint 3 presentation

Road Blocks:

- Need to finish connecting front-end to database

To Do:

- Front-end and database connection
- Create and finalize poster and final presentation

WSR 11:

Weekly Status Report

Project: Butler Theater

Team Members: Eromo Aligbe, Maya Grandstaff, Kameron Leisure,
Geraldyn Miller, and Sam Royal

Week Of: December 9 – 13

Completed:

- Connecting database to front-end application

Road Blocks:

- Cannot figure out how to run queries

To Do:

- Work on how to run queries between front-end and database
- Finish final report and poster

Chapter 8 – Future Work

Since this semester the focus was on creating a strong foundation, there is a lot of work that needs to be accomplished in the future. This includes but is not limited to:

- Proving the connection between the front-end and database
- Getting the front-end to be able to run queries
- Adding in table information to the front-end application
- Adding in remaining tables to the database
- Inputting garment information into the database
- Testing and quality assurance measures for program optimization

There is also room for additional functionality and requirements that can be implemented by the next team. One of the largest ones that can be done is creating a single-page application aside from the administrative page that can serve the sole purpose of running queries about garment information. This would keep the editing and adding of new information separate from running queries and could be an extra security measure so that data does not get messed up.

Along with this, there could be functionality put in place to give an option to add in garment images or even add in barcodes so that items can be scanned and their information looked up from the database.

The Butler Theatre group enjoyed working on this project and hope that there will be further work on this by future groups of the EPICS program. Thank you to Megan Wiegand of the Butler Theatre department for working with us, and to the EPICS program for giving us this opportunity to create a real-world project and further our learning outside of the classroom.

References:

Websites used for help:

[Github.butler.edu](https://github.com/butler-edu)

<https://docs.djangoproject.com/en/2.2/intro/tutorial03/>

Other various websites were used throughout the semester, but a log was not kept as many were used for only one small piece of help

People that were used for reference throughout the semester:

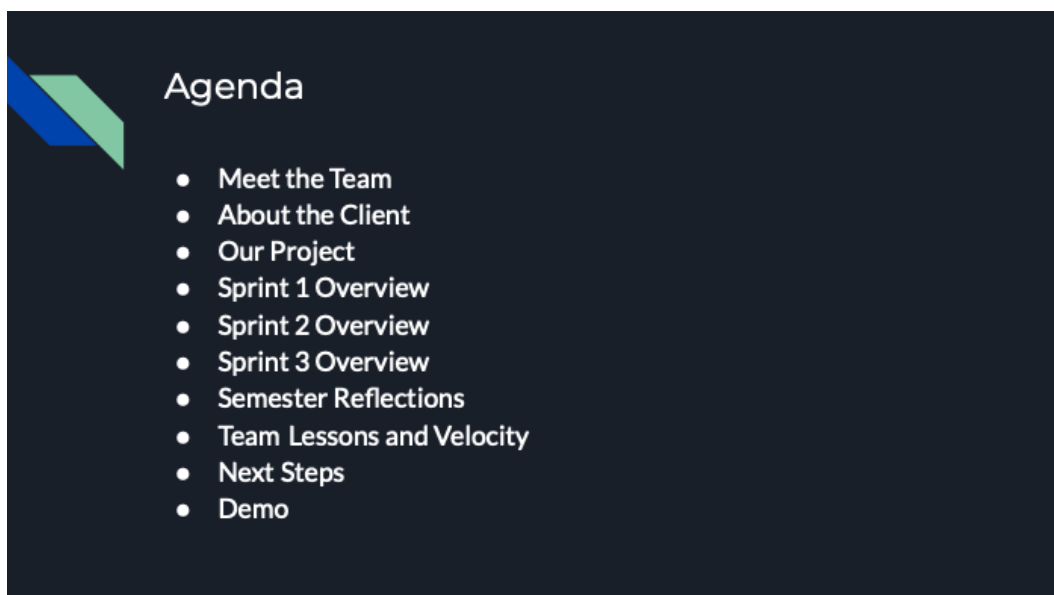
Panos Linos

Troy Wiegand

Nathan Partenheimer

Appendices:

Presentation Slides:





Meet the Team

Eromo Aligbe - Technical Lead

Maya Grandstaff - Team Lead

Kameron Leisure - Database Technician

Geralyn Miller - Documentation and Website Lead

Sam Royal - Client Liaison



About our Client

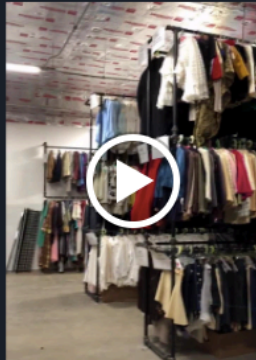
Renowned as one of the best programs in the country, Butler Theatre strives to push the boundaries of our art as we create innovative productions and instill in our graduates an entrepreneurial spirit that will help them to succeed in an ever-changing market. With a strong liberal arts foundation, we foster the development of well-rounded, critical-thinking artists who can plot their own successful path in theatre and in life.



Client's Pain Points

- Items are organized in the warehouse, but there is no record as to which items are actually there
- Need a way to catalogue item information (type, size, color, location, etc.) and a method to be able to access this information
- Goal: Help the theatre department be more efficient and have a better understanding of all of the items that they have

Videos of Warehouse



Our Solution/Project

- Two main components:
 1. Database to store all of the information
 2. Front-end application in order to access the information in the database
- Our Plan this Semester:
 - Set a strong foundation for the database and application
 - Broken up into three sprint projects

Sprint Project Overviews



Sprint 1 Overview

- Meet with client
- Set up EPICS website
- Contact Butler IT about program compatibility
- Pick framework for project
- Create database schema



Sprint 1 Reflections




- Meet with client ✓
- Set up EPICS website ✓
- Contact Butler IT about program compatibility ✓
- Pick framework for project 50%
- Create database schema 50%



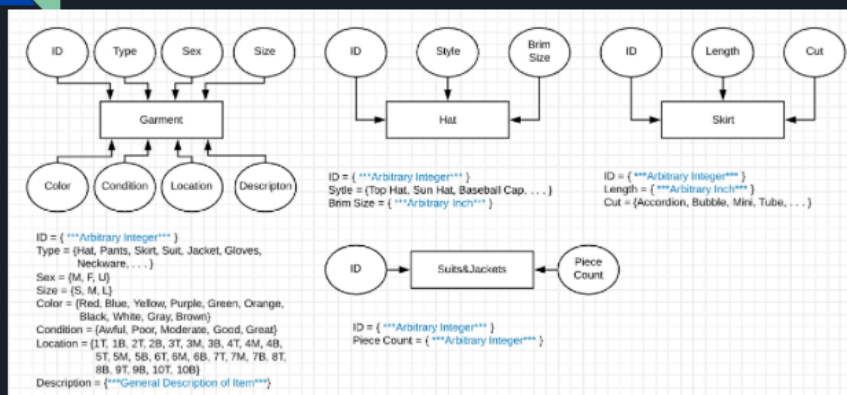
Sprint 2 Overview

- Finalize schema
- Arrange database for schema
- Create prototype of rest API

Sprint 2 Reflections

- Finalize schema 
- Arrange database for schema 
- Create prototype of rest API 

Entity Relationship Diagram



Sprint 3 Overview

- Decide which program to use and execute from this
- Get MySQL server up
- Get website to run queries
- Create poster for end of semester



Sprint 3 Reflections

- Decide which program to use and execute from this ✓
- Get MySQL server up ✓
- Get website to run queries 50%
- Create poster for end of semester ✓




Overall Reflections

- Outlook changed as time went on this semester
 - Went from being extremely ambitious to much more feasible
- Learning new languages and programs can be tough - important to keep trying and to not give up
- Creating a strong foundation is important to carry this project forward for the future



Team Lessons and Velocity

- Take notes to remind ourselves specifically what we are doing/useful code
 - Speeds up the process & increases velocity
- Sometimes you just have to learn as you go, and that's the best way to make visible progress
- It's okay to take a step back and try different methods- it is all a learning process!




Next Steps - future ideas

- Adding in more tables to database
- Editing front-end application
- Making sure that connection between ends is working and is able to run queries
- Start inputting garment information from Butler Theatre collection



Demo!



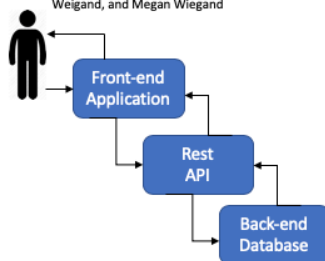
Thank you!
Questions?

Final Poster:

EPICS: Butler Theatre



Acknowledgements:
Nate Partenheimer, Panos Linos, Troy Weigand, and Megan Wiegand



Maya Grandstaff - Team Lead
Eromo Aligbe - Technical Lead
Kameron Leisure - Database Technician
Geralyn Miller - Documentation and Website Lead
Sam Royal - Client Liaison

Team Objectives:

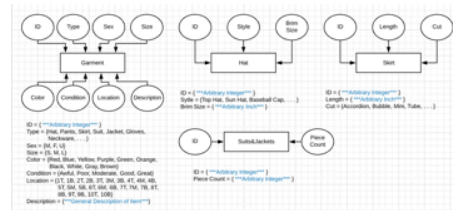
- Establish a schema for the database
- Begin building both front and back-end components of project
- Connect database to front-end application

The Client:

Renowned as one of the best programs in the country, Butler Theatre strives to push the boundaries of our art as we create innovative productions and instill in our graduates an entrepreneurial spirit that will help them to succeed in an ever-changing market. With a strong liberal arts foundation, we foster the development of well-rounded, critical-thinking artists who can plot their own successful path in theatre and in life.

Project Description:

Create a database and front-end application that will hold all of the clothing items that the theatre department has, along with a way that they can access this information.



Reflection:

This semester was a big learning experience for our team, as we learned how essential it is to have a strong base in order to execute the plan properly. While we did not accomplish our lofty goals from the beginning of the semester we did leave good fundamentals for the project going forward.

Source Code – For Front-End:

```

cd ~/projects
mkdir epics && cd epics
python3.6 -m venv venv
source ./venv/bin/activate
pip install Django
python -m django --version
django-admin startproject theatre
cd theatre/
vim theatre/settings.py # add 'thomas.butler.edu' to ALLOWED_HOSTS
python manage.py runserver 0:8000
cd ~/projects/epics/theatre
python manage.py startapp garments
cd ~/projects/epics/theatre/garments/
vim views.py
vim urls.py
cd ~/projects/epics/theatre/theatre/
vim urls.py
vim settings.py
cd ~/projects/epics/theatre
cd ~/projects/epics/theatre
python manage.py migrate
cd ~/projects/epics/theatre/garments/
vim models.py
vim ~/projects/epics/theatre/theatre/settings.py
cd ~/projects/epics/theatre
python manage.py makemigrations garments
python manage.py sqlmigrate garments 0001
python manage.py migrate
  
```

```
cd ~/projects/epics/theatre/garments/  
cd ~/projects/epics/theatre  
python manage.py createsuperuser  
vim ~/projects/epics/theatre/garments/admin.py  
python manage.py runserver 0:8000  
sqlite3 ~/projects/epics/theatre/db.sqlite3
```

Vim portion changes:

Theatre\items\Urls.py:

```
from django.urls import path  
from . import views  
urlpatterns = [  
    path("", views.index, name='index'),  
]
```

Theatre\theatre\urls.py:

```
from django.contrib import admin  
from django.urls import include, path  
urlpatterns = [  
    path('garments/', include('garments.urls')),  
    path('Items/', include('Items.urls')),  
    path('admin/', admin.site.urls),  
]
```

Theatre\garments\models.py:

```
from django.db import models  
# Create your models here.  
class Garments(models.Model):  
    type_text = models.CharField(max_length=200)  
    sex_text = models.CharField(max_length=200)  
    size_text = models.CharField(max_length=200)  
    color_text = models.CharField(max_length=200)  
    condition_text = models.CharField(max_length=200)  
    location_text = models.CharField(max_length=200)
```

Theatre\items\models.py

```
from django.db import models  
# Create your models here.  
class Garments(models.Model):  
    type_text = models.CharField(max_length=200)  
    sex_text = models.CharField(max_length=200)  
    size_text = models.CharField(max_length=200)  
    color_text = models.CharField(max_length=200)  
    condition_text = models.CharField(max_length=200)  
    location_text = models.CharField(max_length=200)  
    pub_date = models.DateTimeField('date acquired')  
class Hats(models.Model):
```

```

Garments = models.ForeignKey(Garments, on_delete=models.CASCADE)
style_text = models.CharField(max_length=200)
brimsize_text = models.CharField(max_length=200)

```

Theatre\theatre\settings.py:

```

import os
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 't=gz+_j0ls&97hs+4r45lq6_2ue4hq8xb^o9&vje+w^be*137q'
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True
ALLOWED_HOSTS = [
    'thomas.butler.edu',
    '127.0.0.1'
]
# Application definition

INSTALLED_APPS = [
    'Items.apps.ItemsConfig',
    'garments.apps.GarmentsConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
ROOT_URLCONF = 'theatre.urls'
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {

```



```

        'context_processors': [
            'django.template.context_processors.debug',
            'django.template.context_processors.request',
            'django.contrib.auth.context_processors.auth',
            'django.contrib.messages.context_processors.messages',
        ],
    },
],
WSGI_APPLICATION = 'theatre.wsgi.application'
# Database
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'theatre',
        'USER': 'BUTLER',
        'PASSWORD': 'theatre',
        'HOST': '10.131.3.172',
        'PORT': '3306'
    }
}
# Password validation
# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
# Internationalization
# https://docs.djangoproject.com/en/2.2/topics/i18n/
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True
# Static files (CSS, JavaScript, Images)

```

```
# https://docs.djangoproject.com/en/2.2/howto/static-files  
STATIC_URL = '/static/'
```