**Progress Report**

**- Increment 2 -**

**Group #2**

1. **Team Members**

Lucas Albano - la14j - LAMango

Jason Santos - jds17e - jds17e

Nick Watts - nsw12b - NickWattsCS

Chris Santos - ces17g - ces17g

Spencer Dennis - sd16g - sd16g

1. **Project Title and Description**

Dance Marathon App.

We are creating an application for the dancers of Dance Marathon to allow them to learn about and check into events, track points and donation amounts, and get notifications by staff.

1. **Accomplishments and overall project status during this increment**

BACKEND:

We have reimplemented the groups module after realizing that it would be simpler to contain information about a user’s organization. As a result, the group module now inherits from Django’s native group class. The User class is now called DanceUser, and instead of inheriting from Django’s Model class, it now extends Django’s native User class. We also decided to consolidate User’s original children classes (Dancer, DanceStaff, and DanceGuest) into DanceUser to make database functionality simpler. This is because Django’s Rest API implementation makes it difficult to implement different child classes and also incorporate the API’s URL to view each child class.

The modification to groups class now holds the organization names, points earned, and funds raised. We’ve changed the attributes of the groups and users classes to reference other database tables through use of ForeignKey and ManyToManyField functions as provided by Django. We have also began working on the graphs module in Django so that we can track and display the amount of money and points raised by each team in the form of a bar graph, pie graph, and leaderboards.

FRONTEND:

The frontend has been doing research on async storage. Additionally, the sign in feature has been fully incorporated as well as the about page has been started. The main focus however was doing research because in order to finish the check-in feature we need to see what user is signed into the application, so when the user checks in, the points for that event can be added to that specific user that is signed in. In order to accomplish this we will be using react native’s async storage library and we are trying to get familiar with how it works.

1. **Challenges, changes in the plan and scope of the project and things that went wrong during this increment**

BACKEND:

One challenge we encountered was that none of us were familiar with Django BUT, since we all knew python we were ready for the challenge! For this increment we have been focusing on educating ourselves with the database aspect of Django, causing some confusion. We have been committing code semi-regularly but with everything that Django has to offer, it can be a little overwhelming but we are at a good pace.

We found a use for the groups class and modified it to consist of organization names, points raised, and the funds raised. Chris was able to figure out a use for the graphs and what to use. The only issue is figuring out where to store data and whether we should have an automation to updating the graphs. Nick, in modifying the User classes to use the Django native user classes, had issues with associated migrations when transferring to the DanceUser class and issues implementing Django’s native group class. The backend team also decided to change the number of classes from four to one due to difficulty getting the children classes to display on the REST API browser extension. This can be changed in the future if we figure out how to display the child class information correctly.

FRONTEND:

The biggest issue this increment has been finishing the sign up functionality. Initially, it was functional but would take 2 clicks to work because of the order the functions were being executed. Also, the profile page was giving an error because it was being updated at the wrong time. It currently switches between the signin in and signed out state properly and effectively. Along with this, there has been issues with the app not working the same on Android and Iphone. Android has some issues with scrolling which might be a result of our navigation techniques.

For events, it was giving errors when checking the code for each event, so we had to rewrite the code that verified users were putting in the correct code. Additionally, we’ve been having to do a lot of research on async storage which is an absolute unit.

1. **Team Member Contribution for this increment**
2. ***Progress Report***

**Lucas**: helped chris with part and 4

**Chris**: wrote (part of 3), (part of 4), part of 6

**Nick**: wrote part of 3 and 4 for backend, and 5 where appropriate

**Jason**: wrote part 4 and helped with 3

**Spencer**: Helped with 3, and 4

1. ***the requirements and design document, including the sections they wrote or contributed to***

**Lucas**: Created the Use Case and Sequence diagrams with online UML tool

**Chris**: Operating Environment,

**Nick**: Updated Functional Requirements

**Jason**: Updated Non-Functional Requirements

**Spencer**: Assumptions and Dependencies, helped with UML diagrams

* 1. ***the implementation and testing document, including the sections they wrote or contributed to***

**Lucas:** Updated 3 for backend

**Chris:** Updated 2 for back end

**Nick:** Updated 5 for backend

**Jason:** Updated parts 4 and 5

**Spencer:** Updated parts 4 and 5

* 1. ***the source code (be detailed about which parts of the system each team member contributed to and how)***

**Lucas**: Still need work on the events model as far as security and other functionality when it comes to querying and retrieve specific info based on certain parameters. Added code that consolidated dance user’s children classes to just dance user. Started work on the images model that will hold information and talk to a data storage that will hold all images. Currently researching code for security.

**Chris**: Wrote additional code to the groups module, working on the graphs portion of the project. Wrote code so groups is now keeping track of organization names, amounts raised, and points raised. Added code for graphs holding data to build on front end. Currently researching code for data visualization.

**Nick**: Extended the DanceUser class (formally the User class) from AbstractUser and consolidated DanceUser’s children classes into just DanceUser through the position attribute. Added implementation to DanceUser to reference other Django database tables. Changed parts of groups module to make migrations to Django’s SQLLite database file. However, there are still errors preventing a full migration.

**Jason**: Created sign up verification, made sure sign up information connects to back end, created log in feature and verification for credentials, started incorporating async storage

**Spencer**: Created events page, UI, getting the back end information of events to connect to front end, check in feature, will be implementing the about page. About page has been started with little code, and have started to attempt to incorporate async storage into the events page.

* 1. ***the video or presentation***

**Lucas**: Talked about adding security features to the events model.

**Chris**: Plans for the next increment

**Nick**: Demonstrating Django challenges and video editing

**Jason**: Demonstration of profile page and successful sign in functionality.

**Spencer**: Demonstration of events page and code/check in verification/about page

**6. Plans for the next increment**

BACKEND:

Creating the charts/graphs for data visualization, and sending them as images to the front end. Incorporating image uploads and storage for them so that they can be posted on front end. Finish debugging users class so that migrations to the database can be made, and figure out how to link the backend’s different database tables to each other.

FRONTEND:

The front end for the next increment hopes to understand async storage and fully incorporate it into our application. As a result, the check-in feature will be fully finished (the last step for check-in just requires async to work to check which user is signed in). Additionally, the about page will be finished and look more pleasing. We also will decide whether or not to keep the sliding when switching tabs (currently users can swipe across tabs, or just select tabs at the bottom, but the swiping can cause issues). Finally, all information will be displayed on the profile screen such as name, position, points, group, teams.

**7. Link to video**

<https://www.youtube.com/watch?v=E4-gRqT12S0>