A3 SRS-1 Team B4

Todd Wegter
Curtis Ullerich
Brandon Maxwell
Yifei Zhu

MINUTES OF MEETING | GROUP # B4 | DATE: 2/10/12

	STUDENT NAME (INITIALS)	Present	Late > 5 mins	Informed about absence?	Scribe?
1	BM	Yes	No	N/A	Yes
2	CU	Yes	No	N/A	No
3	TW	No	N/A	Yes	No
4	YZ	Yes	No	N/A	No

	STUDENT NAME (INITIALS)	OLD ACTION ITEM	STATUS
1	BM	Work on Google Apps	Continuous
2	CU	Work on Html5	Continuous
3	TW	N/A	N/A
4	YZ	Work on Google Apps	Continuous

AGENDA/DISCUSSION SUMMARY

This meeting was treated as an individual work time:

-First we discussed how to represent dates/times on for the assignment

-Curtis continued working on the Html5 stand alone app and made some great progress including adding the ability to store tardy students locally and to edit the students that were counted tardy.

-Fei continued working on the Google Apps tutorials to learn about using the datastore and entities

-Brandon worked on setting up servlets for the website app and read through tutorial code to find out what they are doing.

	STUDENT NAME (INITIALS)	NEW ACTION ITEM	DUE DATE
1	BM	Set up a basic servlet	2/18/12
		Write methods to add students	2/18/12
2	CU	N/A	N/A
3	TW	N/A	N/A
4	YZ	N/A	N/A

MINUTES OF MEETING | GROUP # B4 | DATE: 2/17/12

	STUDENT NAME (INITIALS)	Present	Late > 5 mins	Informed about absence?	Scribe?
1	BM	Yes	No	N/A	Yes
2	CU	Yes	No	N/A	No
3	TW	Yes	No	N/A	No
4	YZ	Yes	No	N/A	No

	STUDENT NAME (INITIALS)	OLD ACTION ITEM	STATUS
1	BM	Set up a basic servlet	Complete
2	CU	N/A	N/A
3	TW	N/A	N/A
4	YZ	N/A	N/A

AGENDA/DISCUSSION SUMMARY

We started out by discussing the SRS1 assignment and divided up the work.

-We decided to put off working on the group parts of the SRS1 assignment until we talked about the assignment more in class.

After talking about the assignment we started working in our sub-groups.

-Todd and Curtis worked together to merge their code.

-Brandon continued working on setting up the database and Fei worked on linking the java and the html parts of the google app engine project.

	STUDENT NAME (INITIALS)	NEW ACTION ITEM	DUE DATE
1	BM	SRS1 Assignment	3/2/12
2	CU	SRS1 Assignment	3/2/12
3	TW	SRS1 Assignment	3/2/12
4	YZ	SRS1 Assignment	3/2/12

MINUTES OF MEETING | GROUP # B4 | DATE: 2/18/12

	STUDENT NAME (INITIALS)	Present	Late > 5 mins	Informed about absence?	Scribe?
1	BM	Yes	No	N/A	Yes
2	CU	Yes	No	N/A	No
3	TW	Yes	No	N/A	No
4	YZ	Yes	No	N/A	No

	STUDENT NAME (INITIALS)	OLD ACTION ITEM	STATUS
1	BM	Write methods to add students	Complete
		SRS1 Assignment	In Progress
2	CU	SRS1 Assignment	In Progress
3	TW	SRS1 Assignment	In Progress
4	YZ	SRS1 Assignment	In Progress

AGENDA/DISCUSSION SUMMARY

Work day:

-Curtis and Todd continued working on the Html5 app

-Brrandon worked more on retrieving and deleting entities from the datastore.

-Fei worked more on the Html/javascript for the webapp

	STUDENT NAME (INITIALS)	NEW ACTION ITEM	DUE DATE
1	BM	None	N/A
2	CU	None	N/A
3	TW	None	N/A
4	YZ	None	N/A

MINUTES OF MEETING | GROUP # B4 | DATE: 2/24/12

	STUDENT NAME (INITIALS)	Present	Late > 5 mins	Informed about absence?	Scribe?
1	BM	Yes	No	N/A	Yes
2	CU	Yes	No	N/A	No
3	TW	Yes	No	N/A	No
4	YZ	Yes	No	N/A	No

	STUDENT NAME (INITIALS)	OLD ACTION ITEM	STATUS
1	BM	SRS1 Assignment	In Progress
2	CU	SRS1 Assignment	In Progress
3	TW	SRS1 Assignment	In Progress
4	YZ	SRS1 Assignment	In Progress

AGENDA/DISCUSSION SUMMARY

At the beginning of the meeting we cleared up some issues that had arrived in the repo. Somehow a bunch of things had gotten deleted, but we were able to go to a previous revision to find it.

After sorting things out with the repo we went back to working in our selective groups.

-Curtis and Todd continued working on the Html5 app

-Brandon worked more on adding, deleting, and retrieving from the datastore

-Fei worked on the Html/Javascript for the webapp

	STUDENT NAME (INITIALS)	NEW ACTION ITEM	DUE DATE
1	BM	N/A	N/A
2	CU	N/A	N/A
3	TW	N/A	N/A
4	YZ	N/A	N/A

MINUTES OF MEETING | GROUP # B4 | DATE: 2/25/12

	STUDENT NAME (INITIALS)	Present	Late > 5 mins	Informed about absence?	Scribe?
1	BM	Yes	No	N/A	Yes
2	CU	Yes	No	N/A	No
3	TW	Yes	No	N/A	No
4	YZ	Yes	No	N/A	No

	STUDENT NAME (INITIALS)	OLD ACTION ITEM	STATUS
1	BM	SRS1 Assignment	In Progress
2	CU	SRS1 Assignment	In Progress
3	TW	SRS1 Assignment	In Progress
4	YZ	SRS1 Assignment	In Progress

AGENDA/DISCUSSION SUMMARY

Today was another work day, like most meetings are.

-Curtis asked Brandon to try to break the html5 app

-Todd worked on refinish the html5 app and finding a background that would be suitable.

-Curtis continued refining his part of the html5 app.

-Brandon continued writing JSPs and setting up the data in the database.

-Fei worked on javascript stuff.

	STUDENT NAME (INITIALS)	NEW ACTION ITEM	DUE DATE
1	BM	N/A	N/A
2	CU	N/A	N/A
3	TW	N/A	N/A
4	YZ	N/A	N/A

MINUTES OF MEETING | GROUP # B4 | DATE: 3/2/12

	STUDENT NAME (INITIALS)	Present	Late > 5 mins	Informed about absence?	Scribe?
1	BM	Yes	No	N/A	Yes
2	CU	Yes	No	N/A	No
3	TW	No	N/A	Yes	No
4	YZ	Yes	No	N/A	No

	STUDENT NAME (INITIALS)	OLD ACTION ITEM	STATUS
1	BM	SRS1 Assignment	Complete
2	CU	SRS1 Assignment	In Progress
3	TW	N/A	N/A
4	YZ	SRS1 Assignment	In Progress

AGENDA/DISCUSSION SUMMARY

Curtis and Fei finished their use case diagrams.

Curtis and Brandon finally had a breakthrough and were able to send the class list to the html5 app.

Fei wrote some of the jsp pages for the website.

	STUDENT NAME (INITIALS)	NEW ACTION ITEM	DUE DATE
1	BM	Implement upload jsp	3/4/12
2	CU	Define class hierarchy	3/4/12
3	TW	Define class hierarchy	3/4/12
4	YZ	Generate more of the JSP	3/4/12

MINUTES OF MEETING | GROUP # B4 | DATE: 3/3/12

	STUDENT NAME (INITIALS)	Present	Late > 5 mins	Informed about absence?	Scribe?
1	BM	Yes	No	N/A	Yes
2	CU	Yes	No	N/A	No
3	TW	Yes	No	N/A	No
4	YZ	Yes	No	N/A	No

	STUDENT NAME (INITIALS)	OLD ACTION ITEM	STATUS
1	BM	Implement upload jsp	In Progress
2	CU	Define class hierarchy	In Progress
3	TW	Define class hierarchy	In Progress
4	YZ	Generate more JSP pages	In Progress

AGENDA/DISCUSSION SUMMARY

We started off this meeting by finishing up the SRS1 assignment.

Brandon worked on setting up some of the class hierarchy and the upload JSP page.

Curtis continued working on the html5 app and interfacing between the server side and the client side.

Todd revised and refined the SRS1 assignment.

Fei continued writing the html on some of the website's JSP pages.

	STUDENT NAME (INITIALS)	NEW ACTION ITEM	DUE DATE
1	BM	Study for Exam!	3/8/12
2	CU	Study for Exam!	3/8/12
3	TW	Study for Exam!	3/8/12
4	YZ	Study for Exam!	3/8/12

Software Requirements Document for B4

Author: Group B4

Brandon Maxwell, Curtis Ullerich, Todd Wegter, Yifei Zhu

Version	Date	Author	Change
0.1	02/17/12	TW	Initial Document
0.2	02/25/12	TW	Completion of all required info (other than Use Cases)
0.3	02/26/12	TW	Addition of textual description of UC1 and UC2
0.4	02/28/12	TW	Addition of sequence diagrams for UC1 and UC2
0.5	03/02/12	BM	Addition of textual descriptions of UC3 and UC4
0.6	03/02/12	YZ	Addition of textual descriptions of UC5 and UC6
0.7	03/02/12	BM	Addition of sequence diagrams of UC3 and UC4
0.8	03/03/12	YZ	Addition of sequence diagrams for UC5 and UC6
0.9	03/03/12	CU	Addition of textual description and sequence diagrams for UC7 and UC8
0.10	03/03/12	TW	Addition of General Use Case Diagram
1.0	03/03/12	TW	First Revision – Ready for proof reading
1.1	03/03/12	CU	Final Edits Made – Ready for Submission
1.2	03/06/12	TW	Final Revisions to Sequence Diagrams – Ready for Resubmission

Table of Contents

1 Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, acronymns, abbreviations 3

1.4 References 3

1.5 Overview 3

2 Overall Description 4

2.1 Product Perspective 4

2.2 Product functions 6

2.3 User characteristics 23

2.4 Constraints 23

2.5 Assumptions and Dependencies 23

3 Specific Requirements 24

3.1 External Interface Requirements 24

3.2 FEATURES 24

3.3 Performance requirements 24

3.4 Design Constraints 24

3.5 Software System Attributes 24

3.6 Other Requirements 24

4 Appendices 25

4.1 Appendix A – Screenflow 25

4.2 Appendix B – ScreenShots 26

1 Introduction

1.1 PURPOSE

This Software Requirements Specification describes the basic functionality of @10dance, an advanced attendance taking system specifically tooled for marching band.

1.2 SCOPE

This document covers eight use cases for @10dance which will showcase its functionality. These use cases will show how the system works and many of the system's capabilities. Non-functional requirements will also be discussed. Screen shots are also included to provide a feel for the system.

1.3 DEFINITIONS, ACRONYMNS, ABBREVIATIONS

Term	Description
Cached	An item which has been cached has been saved to localStorage in the html5 app, but has not yet been uploaded to the server
Static App	This is the server-hosted app and the main portal for interacting with the system
Mobile App	This is an html5 web app designed to work offline and load attendance data to the server when reconnected to the internet.

1.4 REFERENCES

No additional references are required.

1.5 OVERVIEW

[OMIT]

2 Overall Description

@10dance is an easy-to-use system for recording attendance, specifically for a marching band. The system can be used to take attendance and calculate grades, making the marching band attendance-taking process much simpler. It also handles many special cases, such as class conflicts and requested absences, so that the directors need not manually keep track of them. @10dance also allows students to view and keep track of their attendance records and grades through an online portal. This system will make taking attendance much less painful for all members of the band.

2.1 PRODUCT PERSPECTIVE

There are only a few related products to our system. The closest match to our system, Attendance Plus, is available from Rediker Software. It allows a teacher to take attendance for his or her class easily, but it does not have many of the marching-band-specific features that our product will have. Attendance Plus does not allow for students to review their attendance, and it does not calculate grades from attendance. Furthermore, all information from Attendance Plus seems to be stored locally, making it hard for multiple people to take attendance of one group as is required in a marching band.

Concept of Operations

The system is comprised of a mobile html5 app, a static web app, and a server. This allows TAs to take attendance offline at rehearsals and performances with the mobile app. Tardies are also recorded at the time of an event by the tardy students themselves on the mobile app, which will be run as one instance on a tablet at each event. Later, the tardy and absence data is uploaded to the server using the mobile app when the TA's device has regained internet connectivity.

This data is compared to known absence requests submitted through the static app. Grades for the students are then determined by the system from the attendance data. Students can log in to the static app to check their grades and attendance, submit absence requests, and solve absence conflicts. Directors have overriding control of the system, so they can add, edit, and delete students,

override grades, send absence conflict problems to students, and approve or deny student absence requests. At the end of the semester, the directors will be able to export the grades to be uploaded to AccessPlus.

Major User Interfaces

Please refer to Appendix A for the screenflow diagram.

2.1.1.1 Example Screenshot and description

Please refer to Appendix B for the proposed screenshots.

Hardware Interfaces

@10dance will require one or more mobile computers (a laptop, tablet, smartphone, or other internet-connected device with an HTML5 compatible browser) for recording absences and tardies on the field. Personal computers will be used to access the server for students, TAs, and directors. The server will be provided by Google App Engine's hosting service.

Software Interfaces

[OMIT] // example: CGI-URL or function signatures etc

Communication Interfaces

[OMIT] // example: modem etc

Memory Constraints

[OMIT] // RAM, and other storage constraints

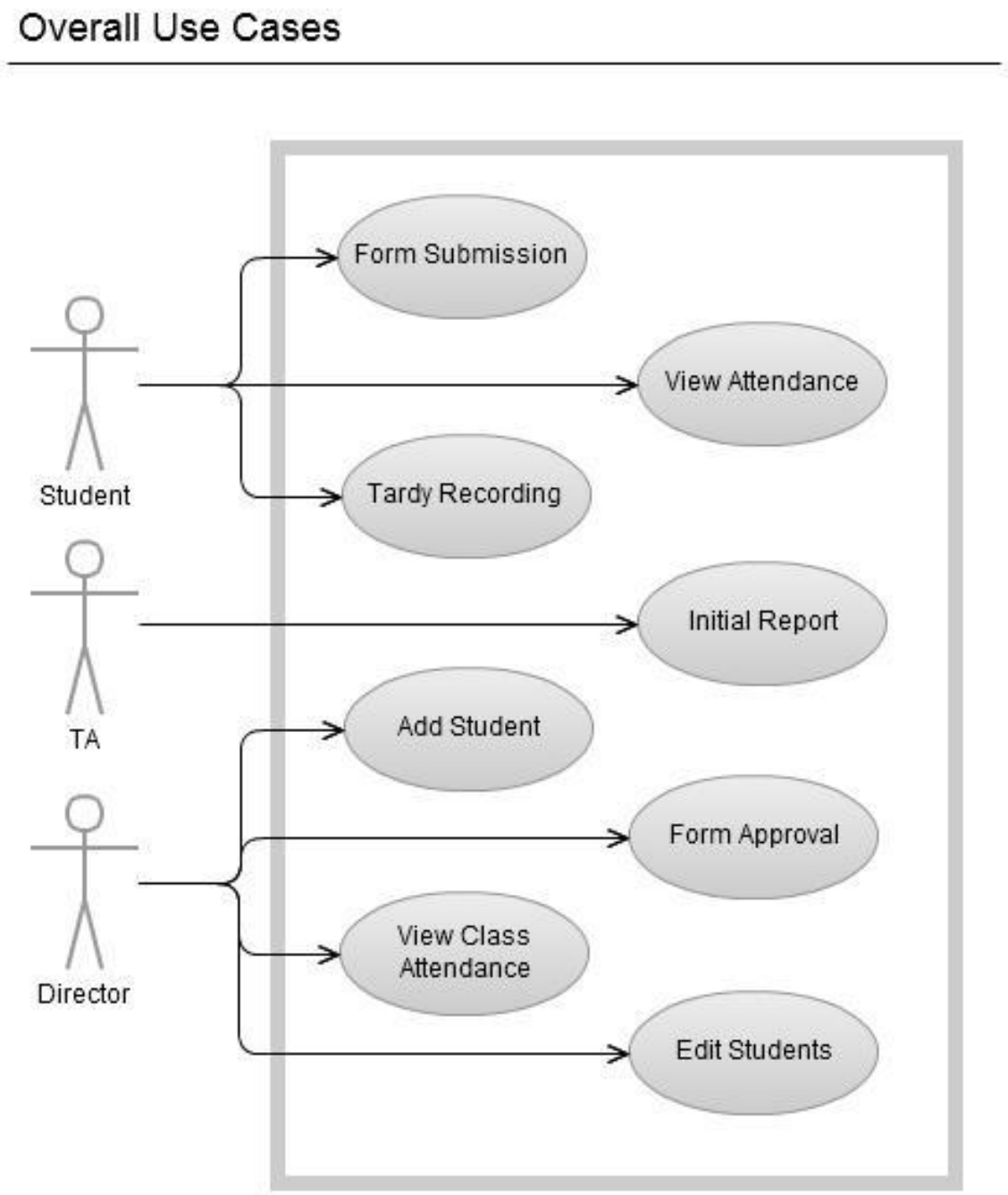
Operations

[OMIT] // special operations (if any)

Site Adaptation Requirements

[OMIT] //ex: Japanese language etc

2.2 PRODUCT FUNCTIONS



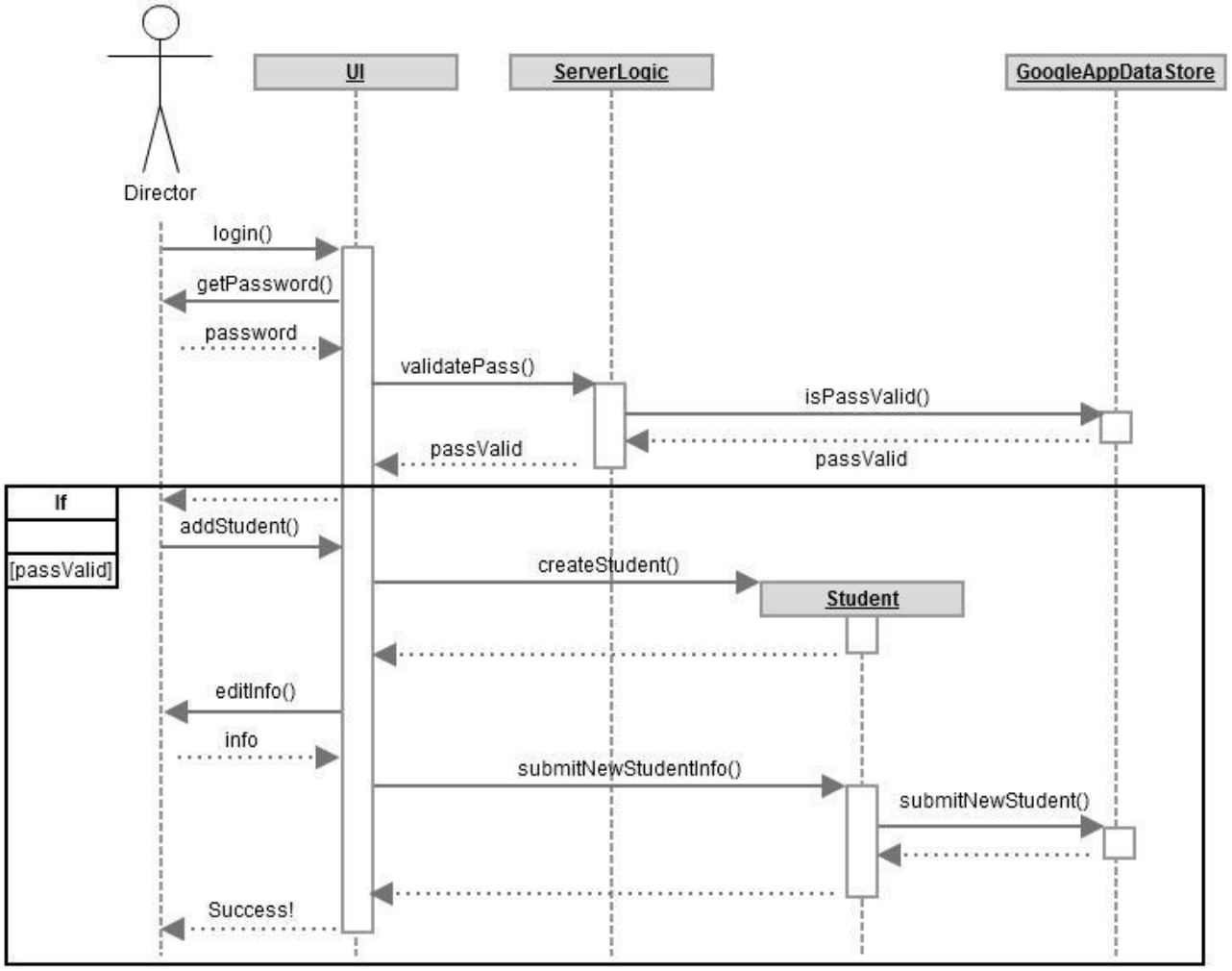
The three actors in our system (student, TA, and director), perform two main classes of actions: Submit information about events or students and edit these items through the online portal. The students and TAs interact with both the static and mobile app, and the director interacts heavily with the static app through the online portal.

2.2.1 – UC1 – Add Student – Todd Wegter

In this use case, the band director is adding a student to the list of students in band. The system returns a confirmation that the student was successfully added upon completion.

a) Actors
    1. The Director
b) Main Flow
    1. The director logs into the system using the Director account
    2. The director navigates to the 'View and Edit Student List' page
    3. The director makes the request to add a student
    4. The system asks the director for the student's info (i.e. name, netID, initial password, etc.)
    5. The director submits the required info.
    6. The system gives confirmation of a successful addition and brings the director back to the 'View and Edit Student List' page, highlighting the newly added student.
c) Alternate Flows
    1. The two password boxes (the original and the confirmation) don't match
        i. An alert is shown asking the director to fix the error
    2. The password contains invalid characters
        i. An alert is shown asking the director to fix the error
    3. A required field is left blank
        i. An alert is shown asking the director to fix the error
    4. The provided netID has already been registered
        i. An alert is shown indicating the error
        ii. The director is taken to that student's existing student page
d) Preconditions
    1. A director account must be established by the developers to allow student list modification.
e) Postconditions
    1. The system will remain in the same state as before, but with one additional student registered in the system, provided that the add operation was a success.

UC1 - Add Student

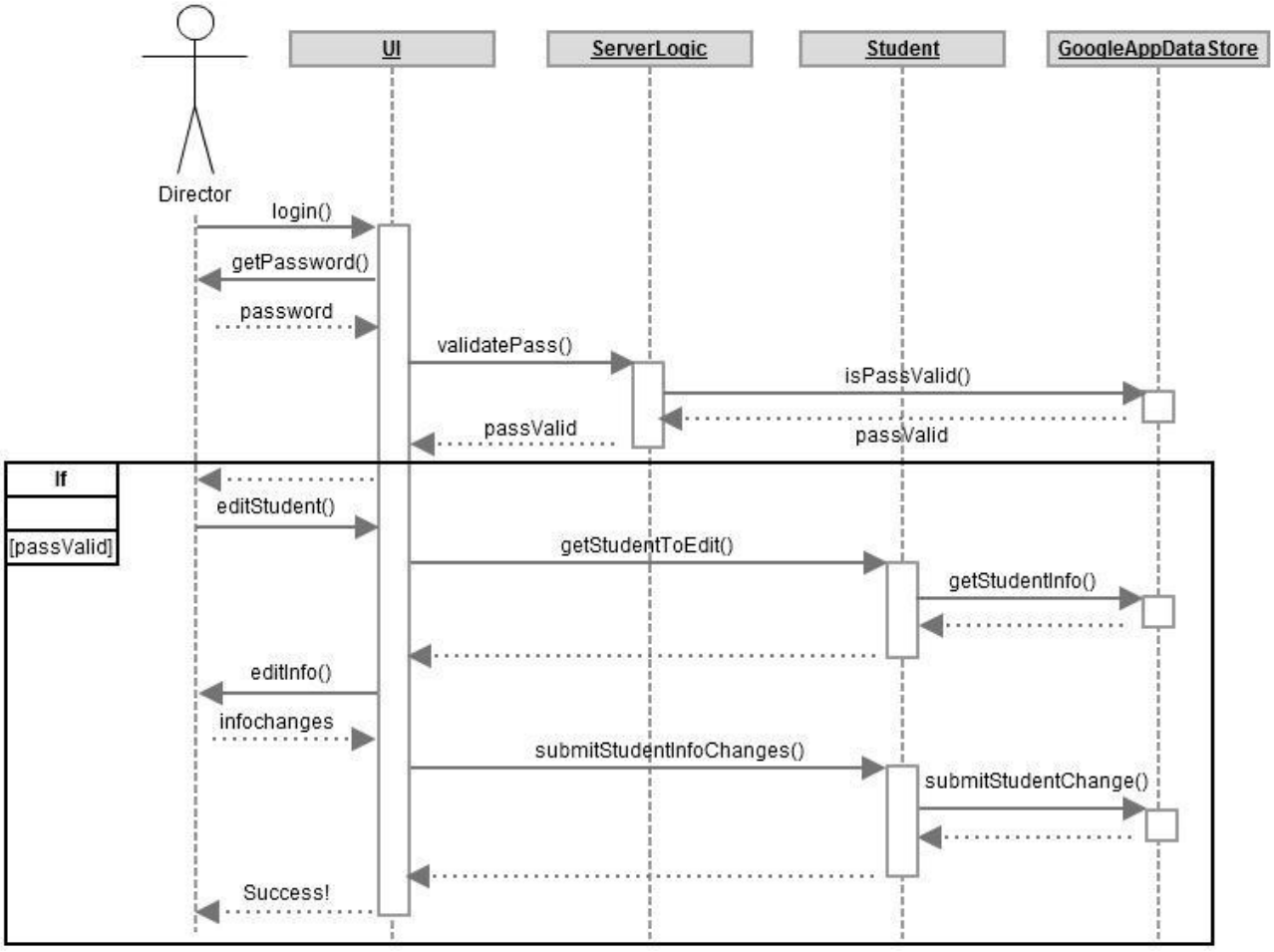


2.2.2 – UC2 – Edit Student Info – Todd Wegter

In this use case, the band director is editing an existing student's info. The system returns a confirmation that the student has been successfully edited.

a) Actors
    1. The Director
b) Main Flow
    1. The director logs into the system using the Director account
    2. The director navigates to the 'View and Edit Student List' page
    3. The director makes a request to view a student
    4. From the student's page, the director makes a request to edit the student.
    5. The system asks the director to change the desired info (i.e. name, instrument, etc.)
    6. The director submits the desired changes.
    7. The system gives confirmation of a successful change a takes the director back to the student's page, where the change can be seen.
c) Alternate Flows
    1. A required field is left blank
        i. An alert is shown asking the director to fix the error
d) Preconditions
    1. A director account must be established by the developers to allow modification of student accounts.
e) Postconditions
    1. The system will remain in the same state as before, but with different information associated with the specified student account, provided that the edit operation was a success.

UC2 - Edit Student

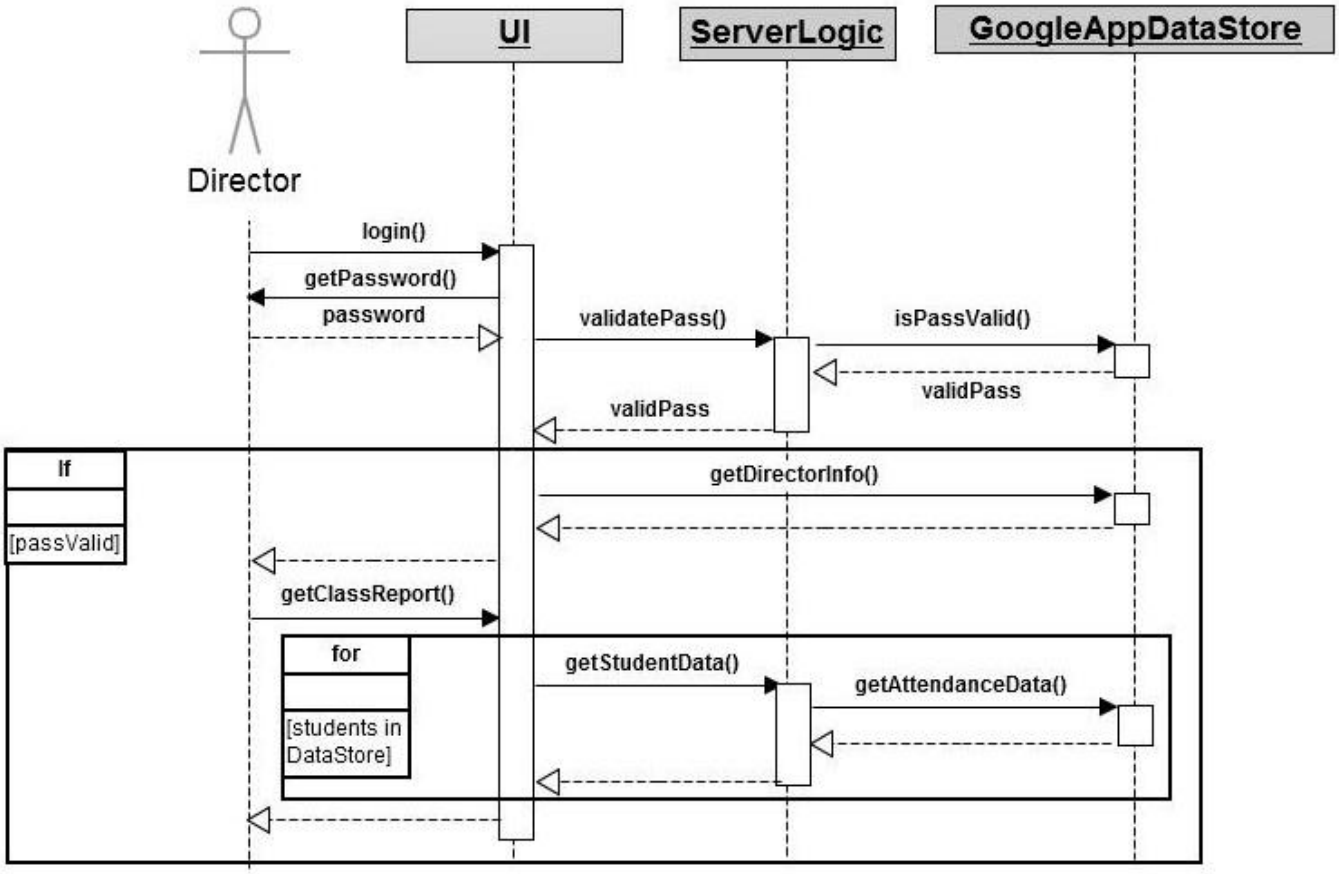


2.2.3 – UC3 – View Class Attendance – Brandon Maxwell

In this use case, the director is using the system to view the aggregated attendance data. The system returns a graph representation of attendance, with the possibility of having this data sorted in different ways.

a) Actors
    1. Director
b) Main flow
    1. The director checks the director checkbox and clicks the next button on the home screen of the web application.
    2. The director logs into the system with a valid netID and password.
    3. The director selects the View Class Reports and Absence Forms button
    4. The system returns a graph of the current attendance for the semester.
    5. The director can optionally change what type of graph, bar, pie, or line, the attendance is viewed as and the system will change to that type of graph.
c) Alternate Flows
    1. The username doesn't exist in the system
        i. An alert appears that lets the user know this mistake has occurred.
    2. The given password doesn't match the password registered for the specific netID provided.
        i. An alert appears to inform the director of this error.
d) Preconditions
    1. There must be a director's account for the user to log into.
e) Postconditions
    1. No information that was in the system has been erased and no new information has been stored.

UC3 - Class Attendance

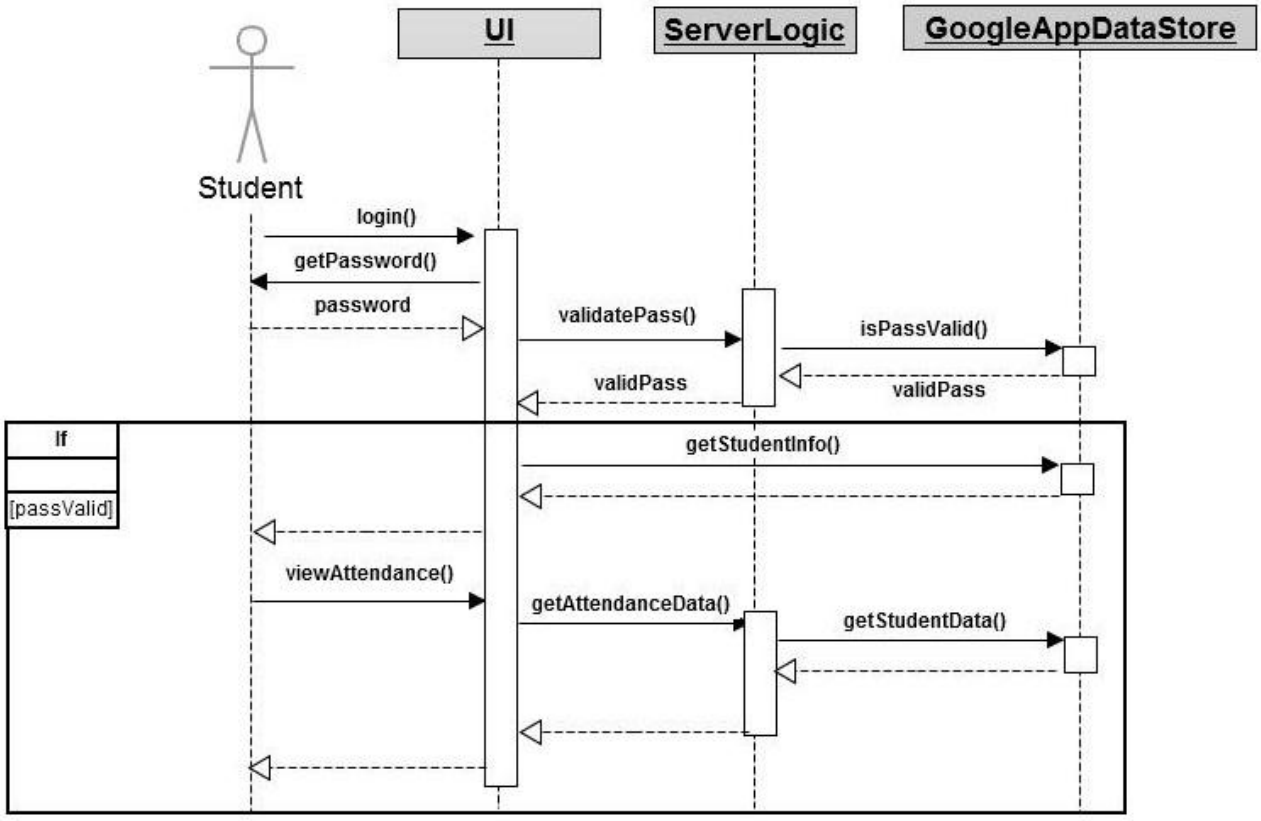


2.2.4 – UC4 – View Individual Attendance – Brandon Maxwell

In this use case, a student is attempting to view their own attendance report. The system returns a graph representation of the percentage of rehearsals/performances the student has attended.

a) Actors
    1. Student
b) Main flow
    1. The student checks the student checkbox and then clicks the next button on the home screen of the web application.
    2. The student uses a correct netID and password to log into the system.
    3. The student clicks the View Attendance button
    4. The system returns a graph of the attendance for that individual.
    5. The student can also click different tabs and the system will return a different form of the same graph. The different graph types are pie, bar, and line.
c) Alternate Flows
    1. The username doesn't exist in the system.
        i. An alert appears that lets the user know this mistake has occurred.
    2. The given password doesn't match the password registered for the netID provided.
        i. An alert appears to inform the student of this error.
d) Preconditions
    1. There must be a student's account for the user to log into.
e) Postconditions
    1. No information that was in the system has been erased and no new information has been stored.

UC4 - View Individual Attendance

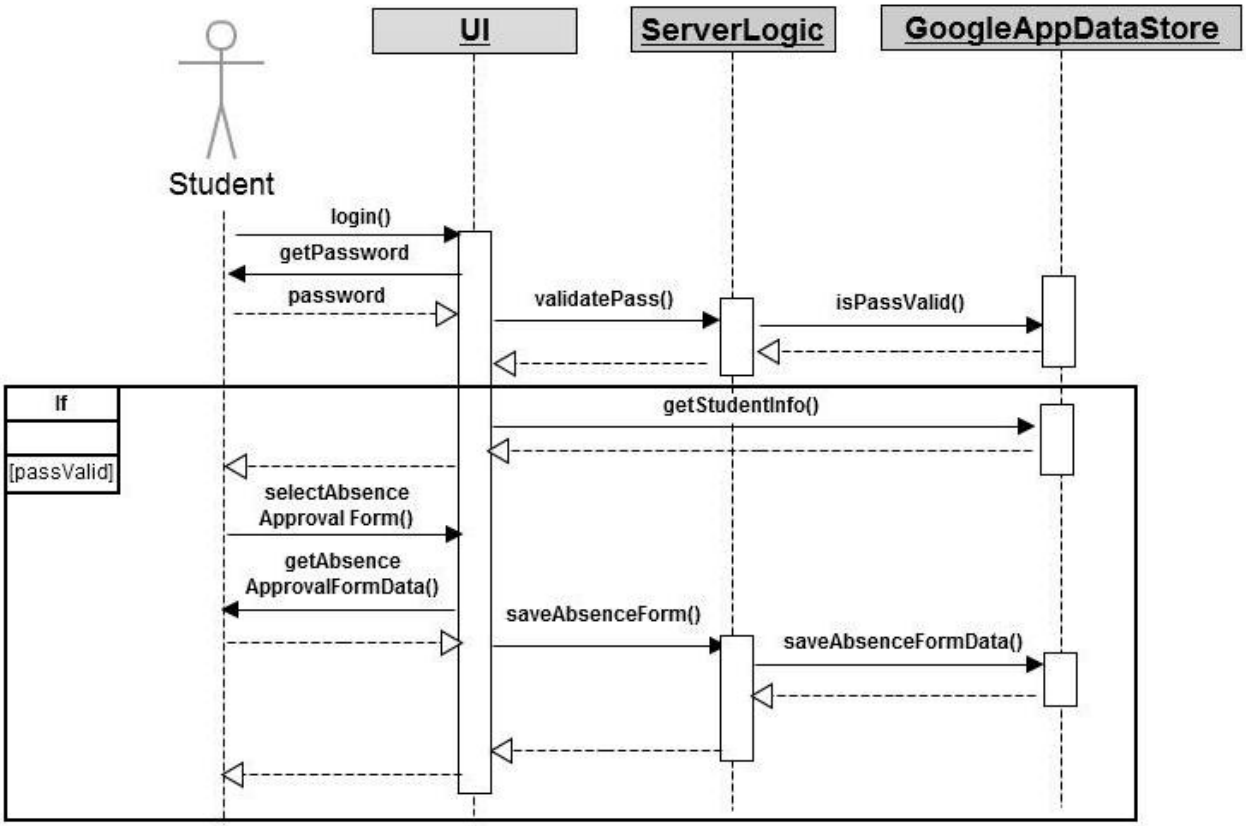


2.2.5 – UC5 – Form Submission – Yifei Zhu

In this use case, a student can submit an Absence Approval Form. These forms alert the system to a student missing a rehearsal or performance ahead of time for either a specific one-time occurrence or a class conflict after director approval. The system returns a confirmation that the successfully submitted the Absence Approval Form.

a) Actors
    1. The Student
b) Main Flow
    1. The student logs into the system using the Student account
    1. The student navigates to the "Absence Approved Form" page
    2. The student selects the Absence Approval Form they want to fill in
    3. The system asks the student for the absence info (i.e. reason, time, date, course info if course conflict, etc.)
    4. The student submits the absence approval form.
    5. The system gives confirmation of a successful submission and brings the student back to the "Student'"page.
c) Alternate Flows
    1. A required field is left blank
        i. An alert is shown asking the student to fill in the appropriate field
d) Preconditions
    1. The student must be enrolled in the class and have an @10dance account to log in and submit the Absence Approval Form.
e) Postconditions
    1. The system will remain in the same state as before, but the submitted form will be sent to director's account, provided that the add operation was a success.

UC5 - Form Submission

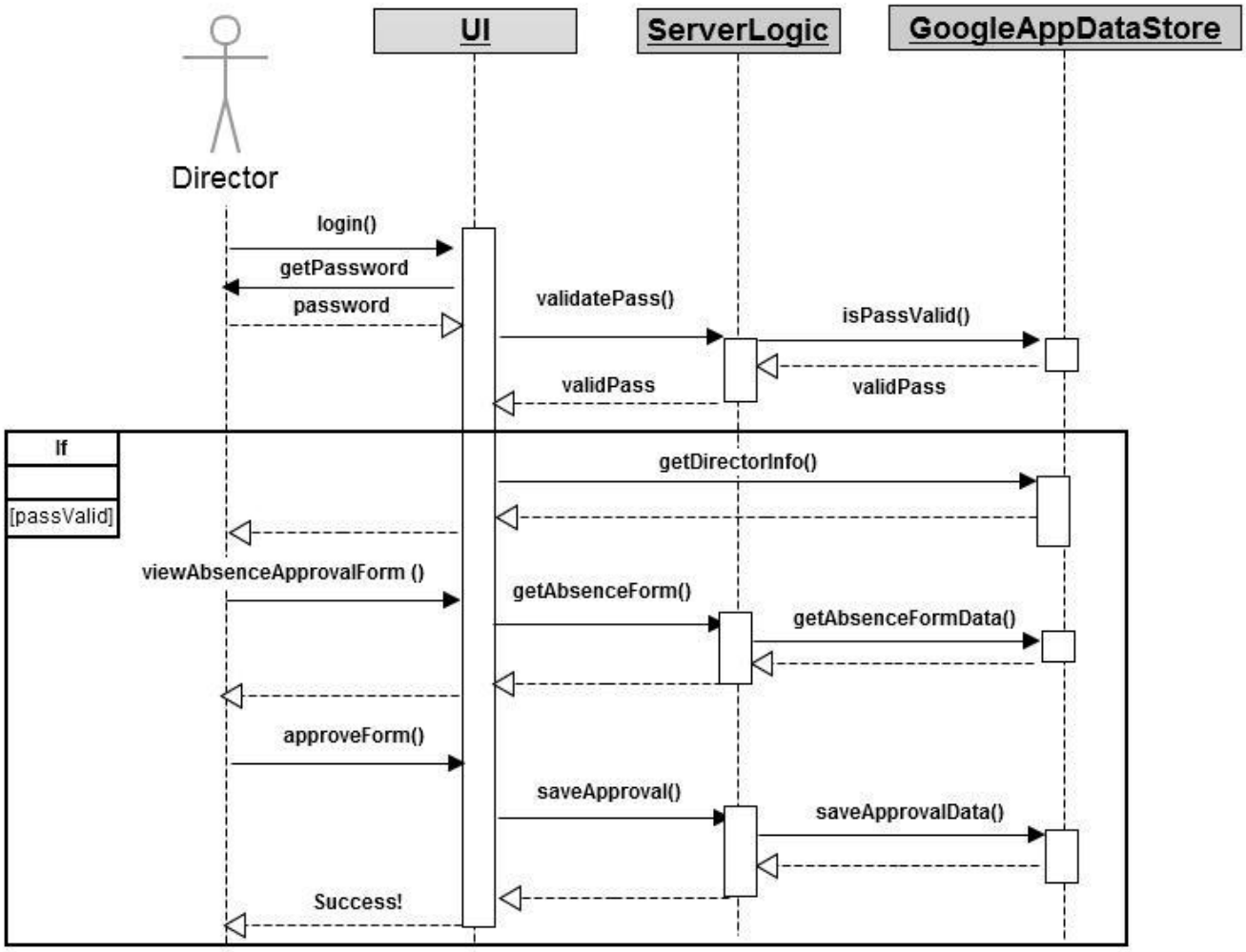


2.2.6 – UC6 – Form Approval – Yifei Zhu

In this use case, the band director is selecting, reading and approving a student's Absence Approval Form in the system. The forms are marked as approved in the band director's systems upon completion.

a) Actors
    1. The Director
b) Main Flow
    1. The director logs into the system using the Director account.
    2. The director navigates to the "View Individual Report" page.
    3. The director brings up the "Absence Forms" section of the "View Individual Report" page.
    4. The director clicks the "View" button to view the student's specific absence approval form.
    5. The director can view details about the absence form.
    6. The director can approve the form by clicking on the "Approve" button or deny the form and leave a message to the student by clicking the "Deny and Message" button.
    7. Upon approval or denial, the system takes the director back to the "View Individual Report" page and marks the director's decision.
c) Alternate Flows
    1. The selected student has not submitted an Absence Approval Form
        i. The "View Individual Report" page shows no unhandled forms, so the director has no forms to look at.
d) Preconditions
    1. A director account must be established by the developers to allow handling of absence forms.
    2. The student must have submitted an Absence Approval Form.
e) Postconditions
    1. The system will remain in the same state as before, but with one additional form approved in the system, provided that the approve operation was a success.

UC6 - Form Approval

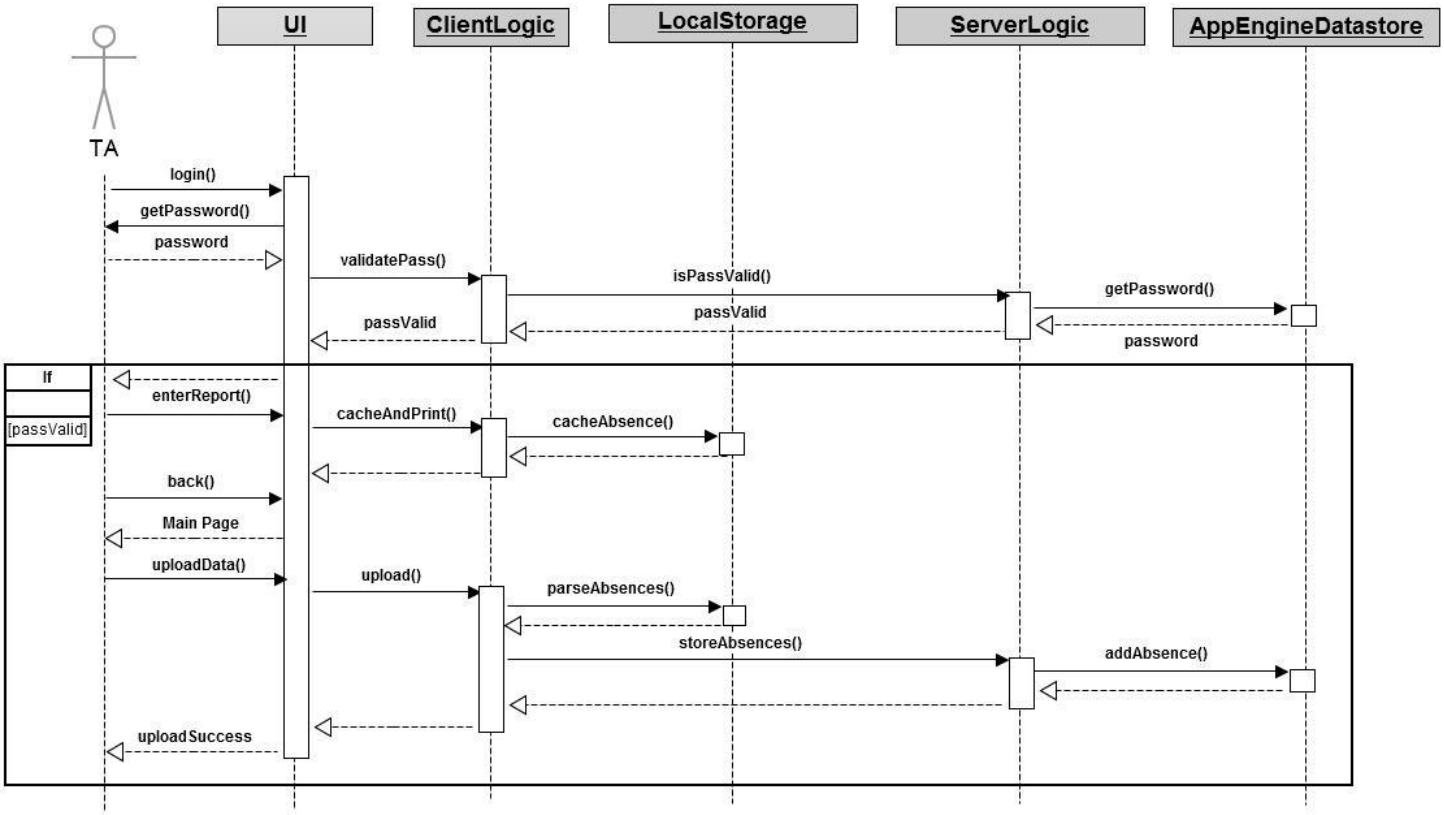


2.2.7 – UC7 – Initial Report – Curtis Ullerich

The TA performs an initial attendance report, noting the details of the current event and the names of all absent students.

a) Actors
    1. TA
b) Main Flow
    1. The TA clicks "Report Absences" from the home screen and enters his or her username and password.
    2. The UI button calls validatePass(), which checks against the hashed password in the database.
    3. The UI proceeds to the initial report screen.
    4. Once on the initial report screen, the TA enters the event information and ticks the name of anyone currently absent.
    5. Once finished, the TA clicks 'submit' and the UI button calls cacheAndPrint(), which calls cacheAbsence() for each entry and prints all entries to the html page.
    6. The TA clicks the back button and the UI returns to the main screen.
    7. The TA clicks the upload button and the UI button calls upload() which calls parseAbsences() to iterate through localStorage and return a list of formatted entries.
    8. The client sends and HTTPRequest to the server which calls addAbsence for each student that has an entry.
    9. Upon successful completion, the UI returns to the home screen.
c) Alternate Flows
    1. The TA credentials entered are incorrect.
        i. The UI displays an error and returns to the main screen.
    2. The client is not connected to the internet.
        i. The UI displays a warning and returns local state to the previous condition.
d) Preconditions
    1. The user has saved the main screen as a bookmark on a device with and HTML5-compatible browser.
e) Postconditions
    1. Any new information is stored in the database.
    2. Device localStorage is empty.

UC7 - Initial Report

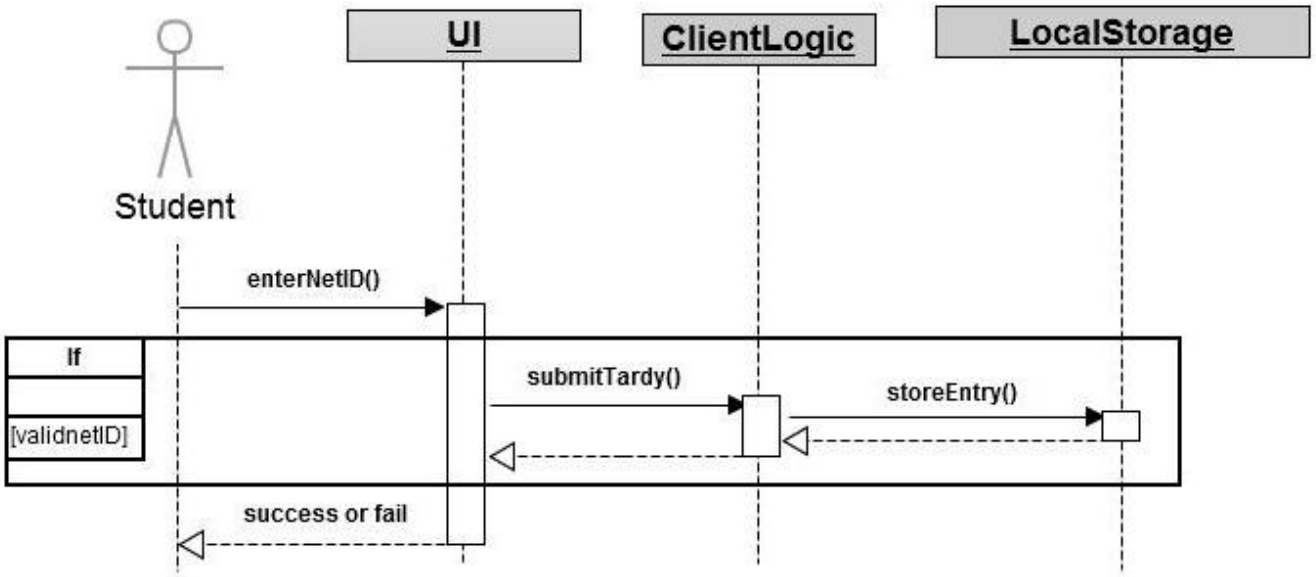


2.2.8 – UC8 – Tardy Recording – Curtis Ullerich

Students check in as tardy on the mobile app in the case where they arrive to rehearsal after the initial attendance report has already been taken.

a) Actors
    1. Student
b) Main Flow
    2. The student enters his or her netID into the text field on the mobile app tardy checkin page.
    3. Hitting 'enter' or clicking 'submit' calls submitTardy() from the UI
    4. The client logic parses the netID and packages it with the current date and time into an entry in localStorage.
    5. The UI returns to the waiting state for the next student submission.
c) Alternate Flows
    1. The entered netID is invalid.
        i. The UI displays an error.
d) Preconditions
    1. The user has saved the main screen as a bookmark on a device with and HTML5-compatible browser.
e) Postconditions
    1. Any new information is stored in localStorage.

UC8 - Tardy Recording



2.3 USER CHARACTERISTICS

The marching band will use this system at least once daily during the marching band season to take attendance with the mobile app. Students will use the static app to monitor and deal with absences as often as they would like. With 350 students in the marching band, the system should be able to handle upwards of 350 logins per day. The directors will also need to log in to the static app several times per day.

2.4 CONSTRAINTS

[OMIT]

// all conditions that may limit design options (INCLUDE NON FUNCTION CONSTRAINTS)

2.5 ASSUMPTIONS AND DEPENDENCIES

[OMIT]

// hardware and software assumptions and dependencies

3 Specific Requirements

[OMIT]

// Here you need to put in details (if any). Mark items [None] if you do not have any information.

3.1 EXTERNAL INTERFACE REQUIREMENTS

User Interfaces

Hardware Interfaces

Software Interfaces

Communications Interfaces

3.2 FEATURES

FEATURE-1 ….

FEATURE-2 ….

FEATURE-3 ….

3.3 PERFORMANCE REQUIREMENTS

3.4 DESIGN CONSTRAINTS

3.5 SOFTWARE SYSTEM ATTRIBUTES

Reliability

Availability

Security

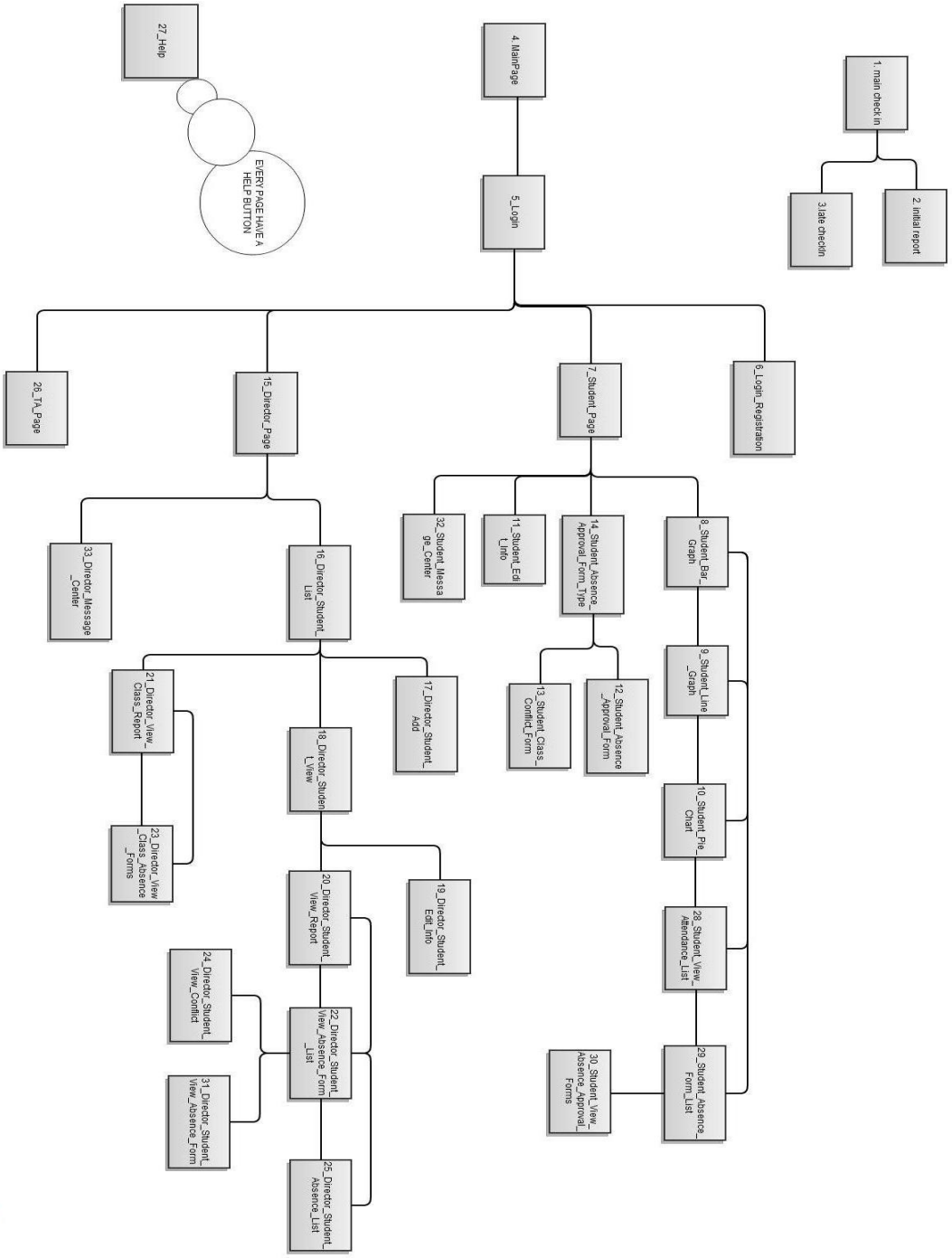
Maintainability

Portability

3.6 OTHER REQUIREMENTS

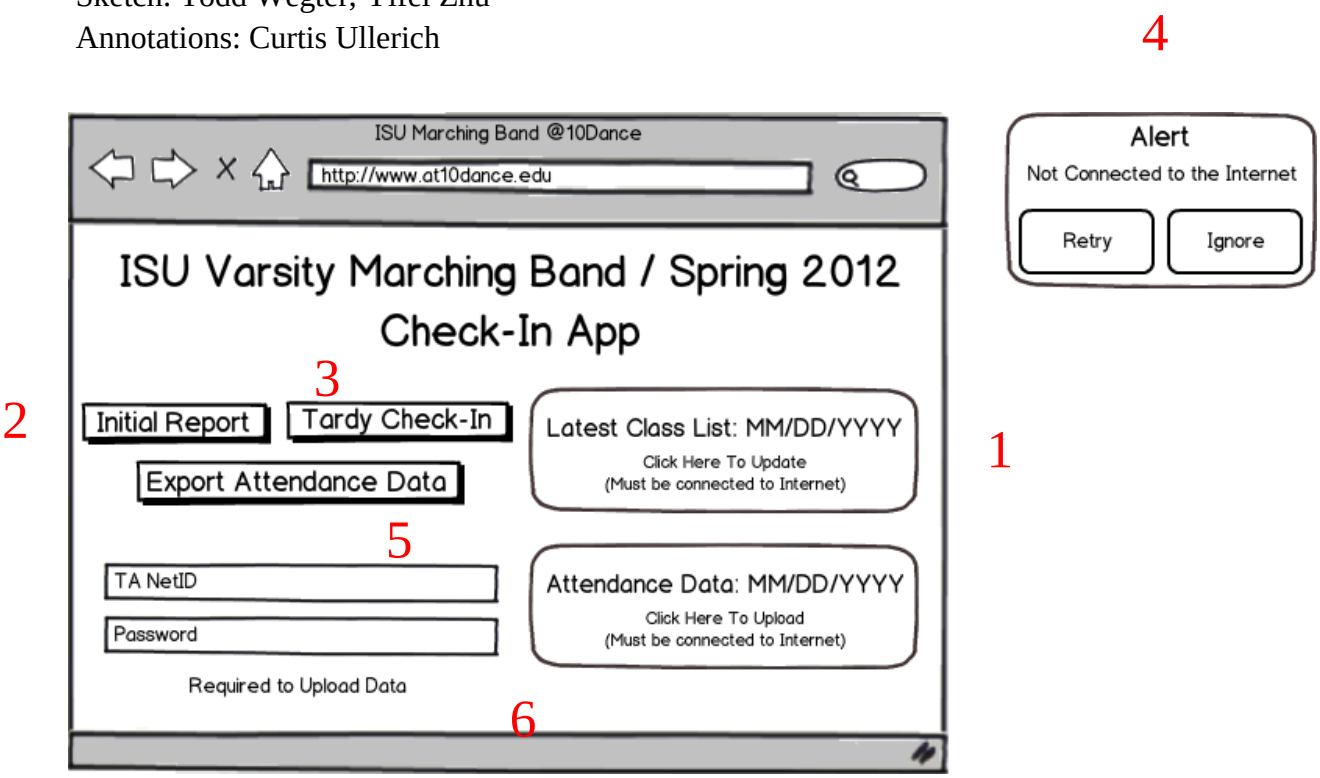
4 Appendices

4.1 APPENDIX A – SCREENFLOW



4.2 APPENDIX B – SCREENSHOTS

1_Main_CheckIn
Sketch: Todd Wegter, Yifei Zhu
Annotations: Curtis Ullerich

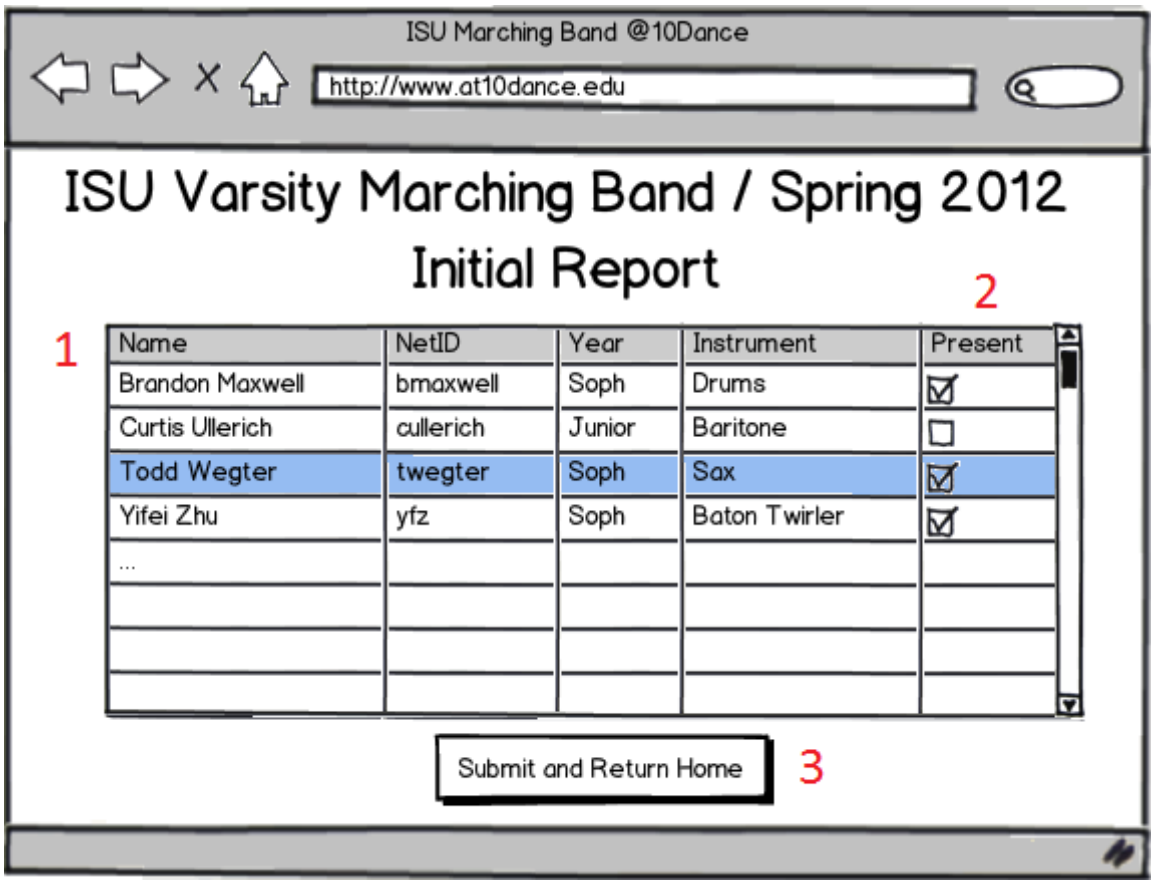


Purpose: Landing screen from which student staff may initiate daily attendance tasks.

Loading Events: User logs in to html5 webapp.

Interface Elements:
1. Latest Class List. Clicking here allows the user to update to the latest class list.
2. Allows the user to create an initial attendance report at (2_Initial_Report) for each rehearsal by selected all absent students.
3. Allows users to check in to rehearsal as tardy at (3_Late_CheckIn).
4. Alert. Clicking (1) or (6) while not connect to a network presents this warning.
5. Exports Attendance data to an uploadable file using 26_TA.
6. Uploads Attendance data to server. A TA's NetID and password must be entered into the appropriate fields to allow uploading.
7. Fields in which TA must enter NetID and password to upload data

2_Initial_Report
Sketch: Todd Wegter, Yifei Zhu
Annotations: Brandon Maxwell

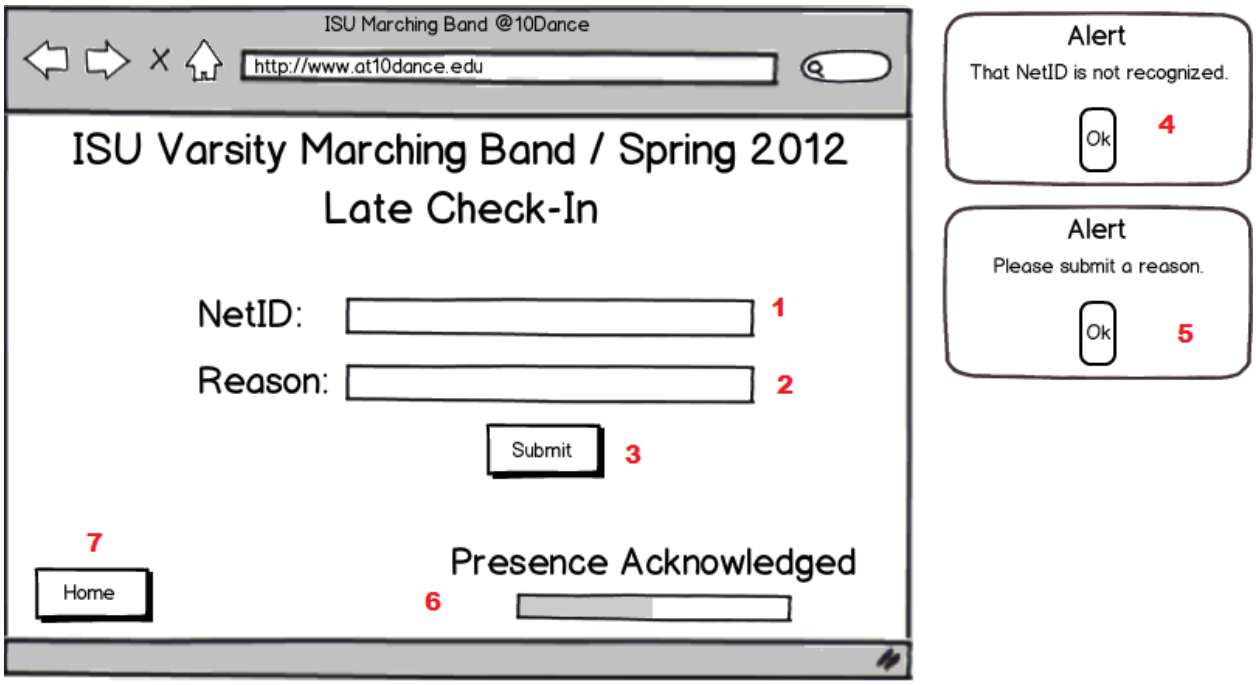


Purpose: To provide a grid view of the students so it is easy for TAs or instructors to mark who is at practice and who is not at practice at the time attendance is taken.

Loading Events: User presses the "Initial Report" button on the home screen

Interface Elements:
1. The entire grid that shows each students' name, NetID, year, and instrument. Used to keep track of what student the TA is marking attendance for.
2. The column of checkboxes. When a student attends practice the TA can check the checkbox and continue onto another student.
3. A button to return the user from this current screen back to the homescreen
(Screen1_Main_CheckIn)

3_Late_CheckIn
Sketch: Todd Wegter, Yifei Zhu
Annotations:Yifei Zhu

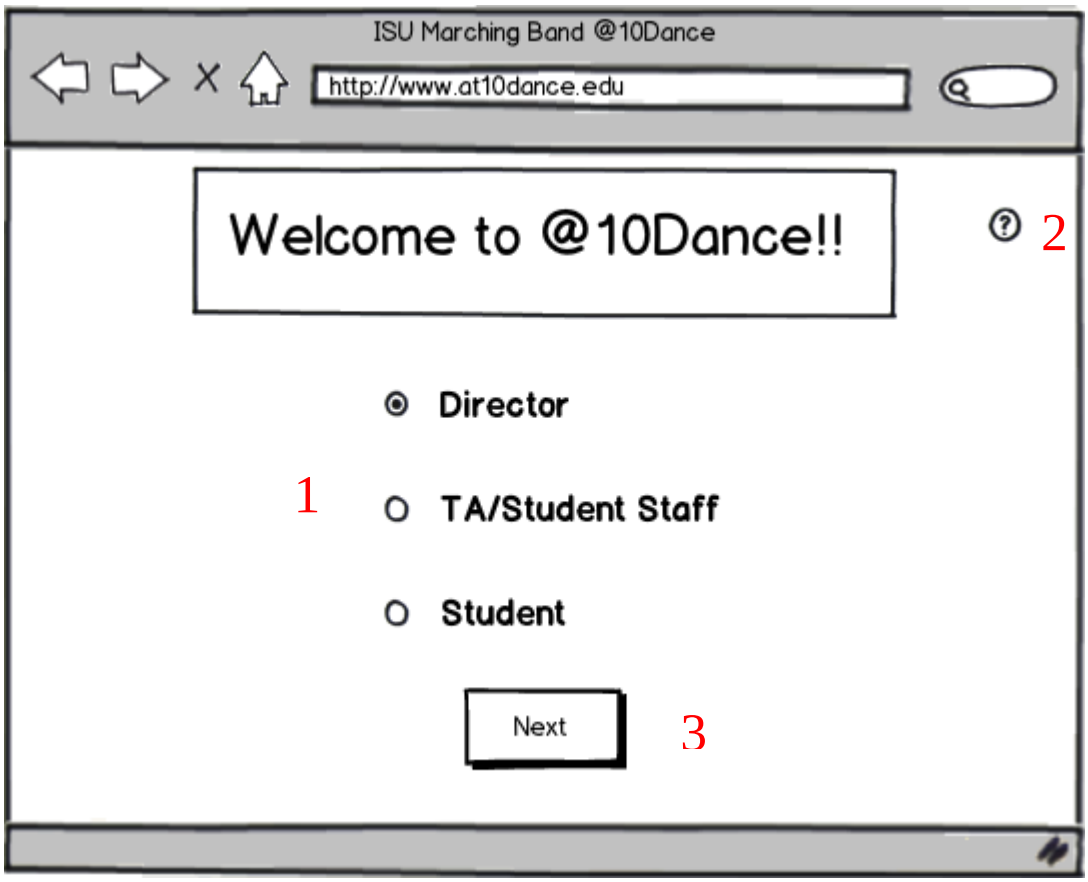


Purpose: Allow late student to check in and explain the reason why is late for Marching Band practice.

Loading Events: User presses "Tardy Check-In" button on the main interface

Interface items:
1. The text field allows users to enter their NetID.
2. The text field allows users to enter the reason why they are late for Marching Band practice.
3. The "submit" buttons allow users to submit their tardy check-in form.
4. If the user enters the wrong NetID or the NetID does not exist, then an Alert will pop up to notify the user that the NetID is incorrect.
5. If the user leaves the "Reason" text field blank, then an alert will pop up and tell the user to enter a reason.
6. The loading bar shows users the progress of submission, after the user clicks on the "Submit" button.
7. The home button will take users back to the main check in homepage (Screen 1_Main_CheckIn)

4_Main_Page
Sketch: Todd Wegter , Yifei Zhu
Annotations: Todd Wegter

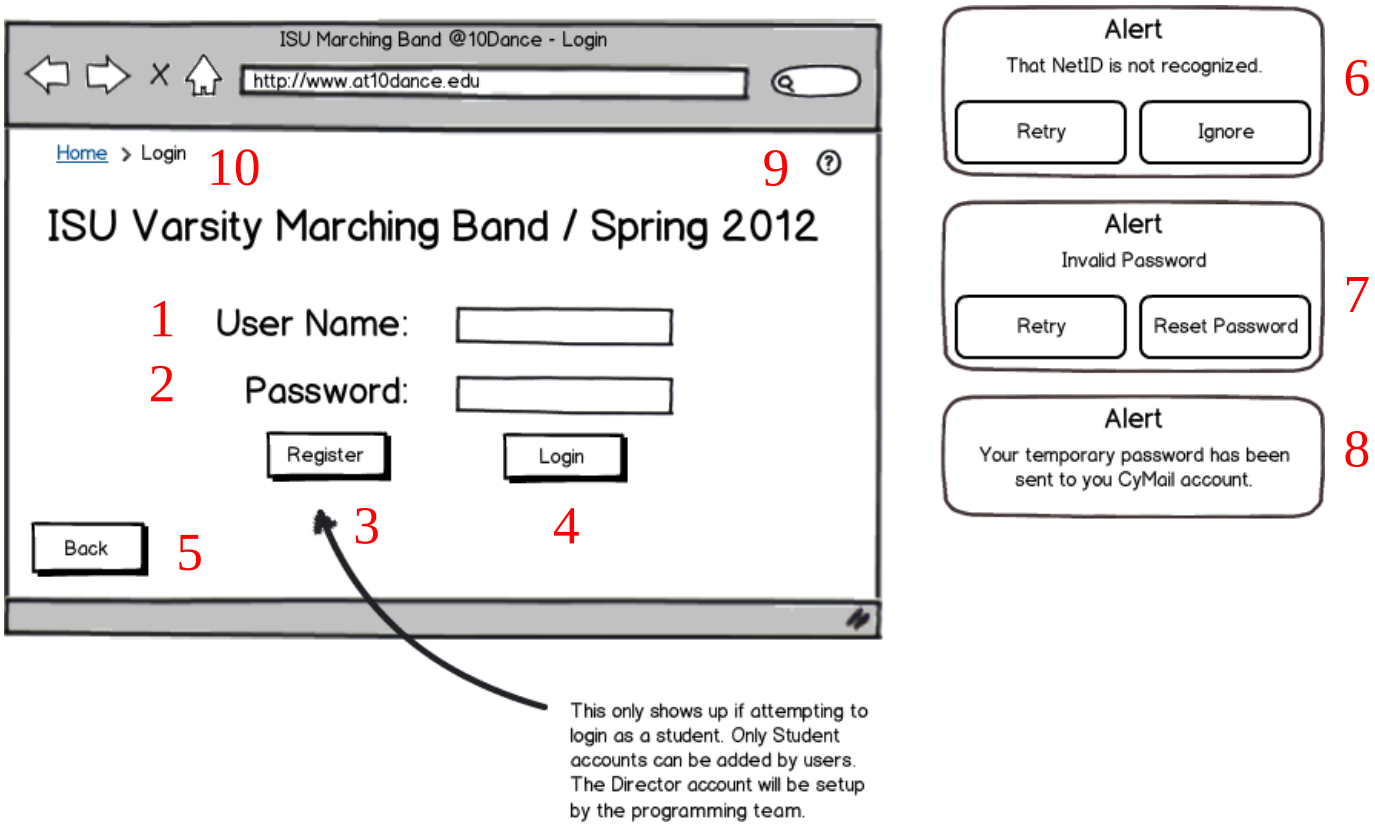


Purpose: To provide an opening screen for the @10Dance application. This web application is the main part of the system. It allows students to check their attendance records, TAs to upload attendance, and the directors to have supreme control over everything.

Loading Events: User opens the @10Dance Web Application

Interface Elements:
1. Radio Buttons allow user to select whether he or she wishes to proceed as a Director, TA, or Student.
2. Help Button pulls up a specialized Help screen (Screen 27_Help) explaining that the user may proceed to the desired login screen by selecting the appropriate radio button (1) and clicking "Next" (3).
3. "Next" Button takes the user to a login screen (Screen 5_Login), which is slightly different for the different user types and allows only the selected type to login.

5_Login
Sketch: Todd Wegter, Yifei Zhu
Annotations: Cutis Ullerich



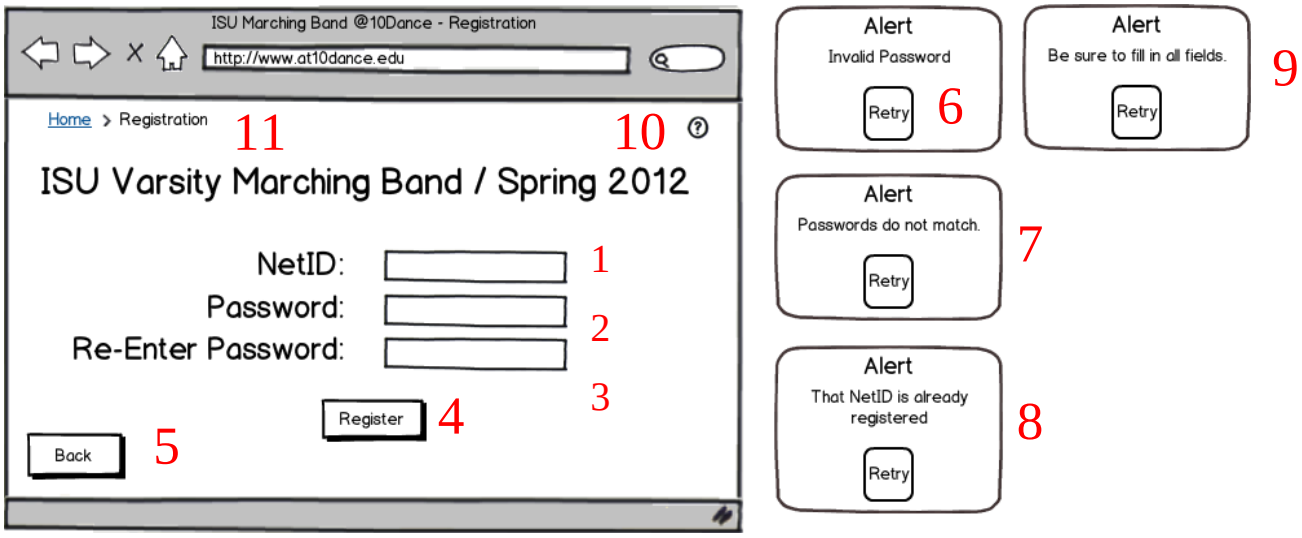
Purpose: To allow users to log in to the system.

Loading Events: User visits the @10dance application URL.

Interface Elements:
1. Field into which user enters his or her NetID.
2. Field into which user enters his or her password.
3. Allows users to register in the system as a student for the current term of marching band.
4. Verifies the user's credentials and allows access to the system.
5. Returns the user to the home screen (4_Main_Page).
6. Alerts the user that the entered NetID is not an entry in the system.
7. Alerts the user that the password does not match the entered NetID. This does not appear if element 6 appears, even though the password is inherently incorrect.
8. Sends a temporary password to the user's CyMail account in the event of a forgotten password.

9. Sends user to a specialized help screen (27_Help) indicating that the user may enter a valid NetID and password combination to log in to the system or register for the system by clicking register (3)
10. Page Trail. This allows the user to go back to the home page (4_Main_Page).

6_Login_Registration
Sketch: Todd Wegter
Annotations: Curtis Ullerich

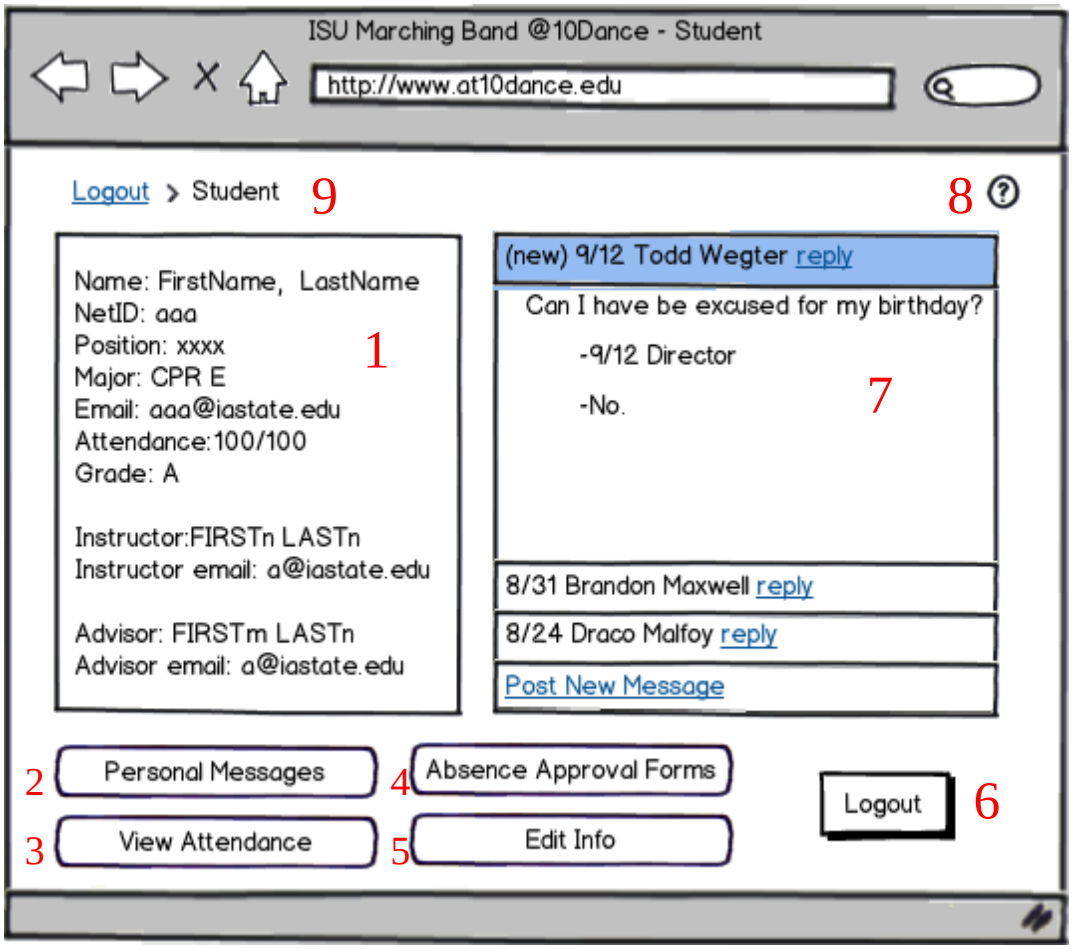


Purpose: To allow students to register for the system for the current term.

Loading Events: User clicks register on the login screen (element 3 on 5_Login)

Interface Elements:
1. The NetID to be registered into the system.
2. The user's desired password.
3. Password confirm field.
4. Returns the user to the login screen (5_Login).
5. Creates an entry for the student in the database.
6. Alerts the user that the password does not conform to system security standards.
7. Alerts the user that the password and password-confirm fields are not identical.
8. Alerts the user that the entered NetID is already in the system.
9. Alerts the user that he or she has left a field empty.
10. Sends user to a specialized help screen (27_Help) indicating that the user may enter a new NetID and valid password to register for the system.
11. Page Trail. This allows the user to go back to the home page (4_Main_Page).

7_Student_Page
Sketch: Todd Wegter, Yifei Zhu
Annotations: Curtis Ullerich



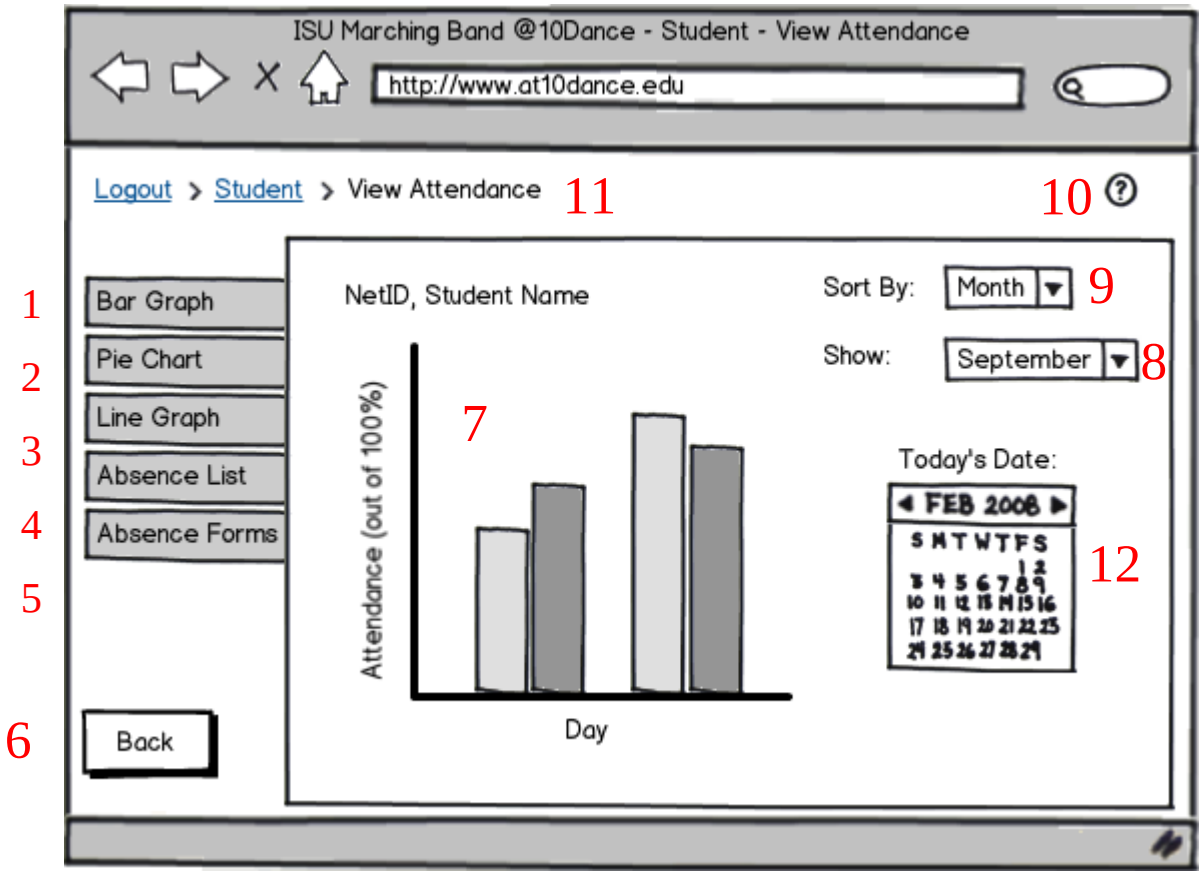
Purpose: To provide a student user an overview of his or her information and access to forms messaging.

Loading Events: Successful login (6_Login_Registration)

Interface Elements:
1. Provides an overview of all student details in the system.
2. Allows the student to view any personal messages sent through the system (screen 32_Student_Message_Center) (a glimpse of which is visible in element 7).
3. Displays the attendance record for this particular student (8_Student_Bar_Graph) from where the student may view other visualizations of his or her data.

4. Take the user to (14_Student_Absence_Approval_Form_Type) from where he or she may submit an attendance form.
5. Allows this student to modify his or her system information (which is displayed in element 1) at screen (11_Student_Edit_Info)
6. Logs the student out of the system and returns him or her to (4_Main_Page)
7. Shows the most recent messages delivered to this student.
8. Sends user to a specialized help screen (27_Help) indicating that the user may view or edit information or submit forms and messages.
9. Page Trail. This allows the user to go back to the home page (4_Main_Page).

8_Student_Bar_Graph
Sketch: Todd Wegter, Yifei Zhu
Annotations: Curtis Ullerich



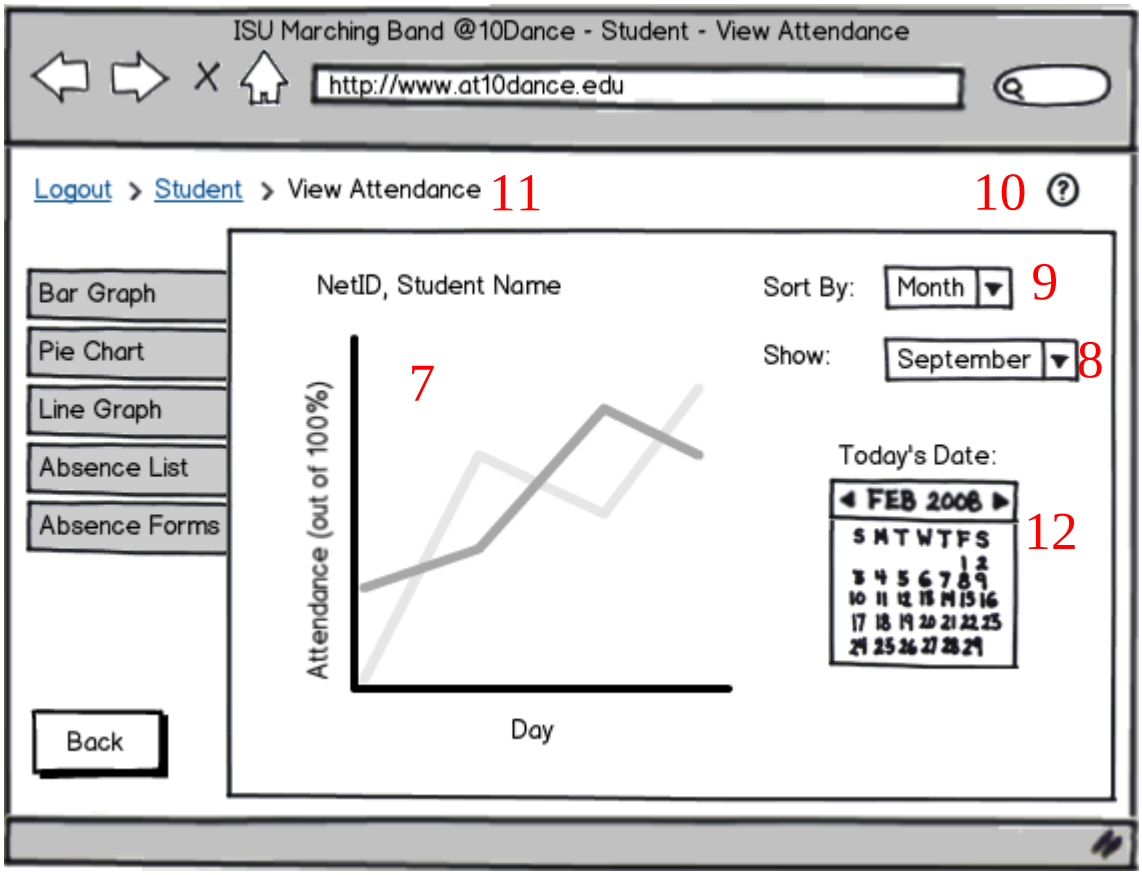
Purpose: To visualize the student's attendance data for the current season to date.

Loading Events: Clicking element 3 on (7_Student_Page).

Interface Elements:

1. Clicking this tab does nothing on this screen, as the current view is already "bar graph."
2. Loads a pie chart visualization of the student's attendance data (10_Student_Pie_Chart)
3. Loads a line graph visualization of the student's attendance data (9_Student_Line_Graph)
4. Loads a list of all absences for this student (28_Student_View_Attendance_List)
5. Loads a list of all absence forms this student has submitted (29_Student_Absence_Form_List)
6. Returns to (7_Student_Page)
7. The bar graph visualization of the student's attendance record based on inputs 7 and 8.
8. Shows data within a selected date range.
9. Sorts the data by various parameters.
10. Sends user to a specialized help screen (27_Help) indicating how the user may view their attendance record and any submitted forms.
11. Page Trail. This allows the user to go back to the student page (7_Student_Page). Logout takes the user back to the home page (4_Main_Page).
12. Allows the user to select the date. Loads as today by default.

9_Student_Line_Graph
Sketch: Todd Wegter, Yifei Zhu
Annotations: Curtis Ullerich



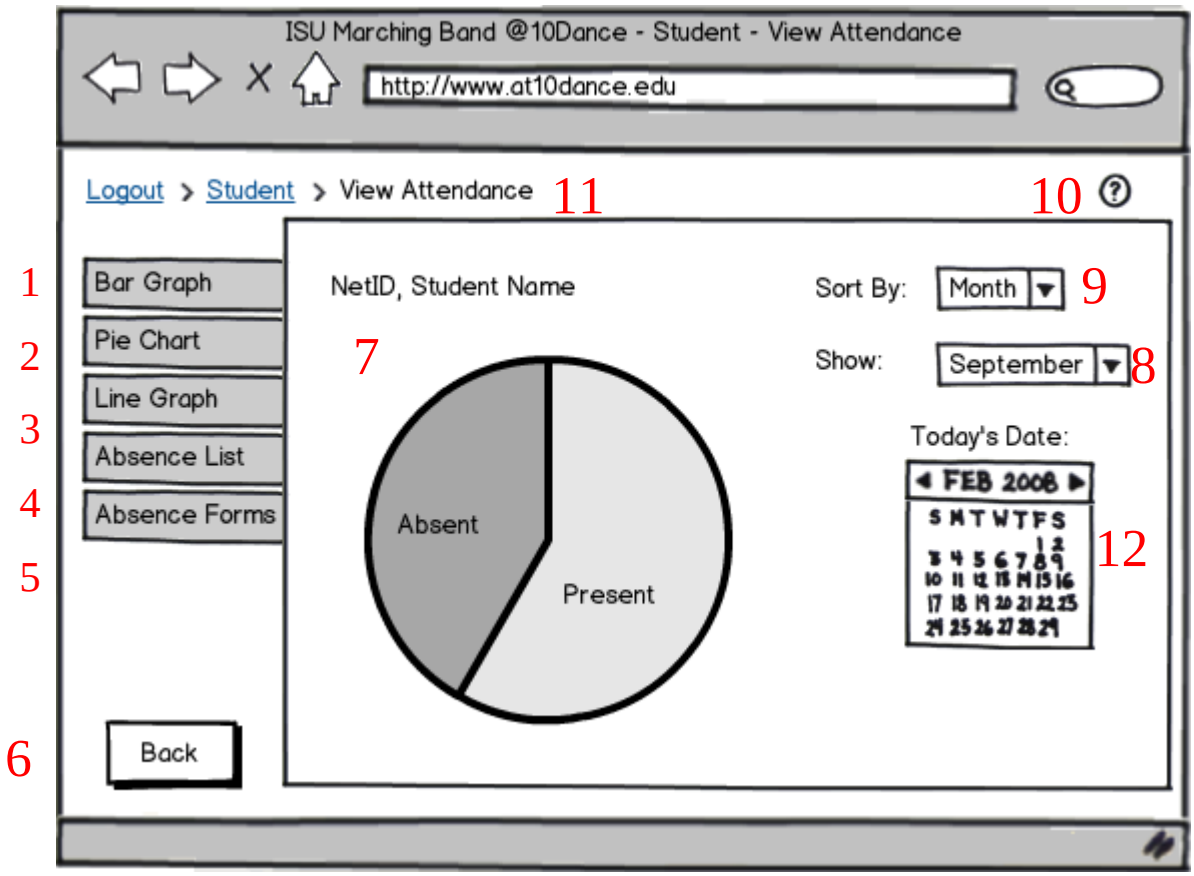
Purpose: To visualize the student's attendance data for the current season to date in a line graph.

Loading Events: Clicking element 3 on (8_Student_Bar_Graph or 10_Student_Pie_Chart)

Interface Elements:

1. Loads a bar graph visualization of the student's attendance data (9_Student_Line_Graph)
2. Loads a pie chart visualization of the student's attendance data (10_Student_Pie_Chart)
3. Clicking this tab does nothing on this screen, as the current view is already "line graph."
4. Loads a list of all absences for this student (28_Student_View_Attendance_List)
5. Loads a list of all absence forms this student has submitted (29_Student_Absence_Form_List)
6. Returns to (7_Student_Page)
7. The bar graph visualization of the student's attendance record based on inputs 7 and 8.
8. Shows data within a selected date range.
9. Sorts the data by various parameters.
10. Sends user to a specialized help screen (27_Help) indicating how the user may view their attendance record and any submitted forms.
11. Page Trail. This allows the user to go back to the student page (7_Student_Page). Logout takes the user back to the home page (4_Main_Page).
12. Allows the user to select the date. Loads as today by default.

10_Student_Pie_Chart
Sketch: Todd Wegter, Yifei Zhu
Annotations: Curtis Ullerich



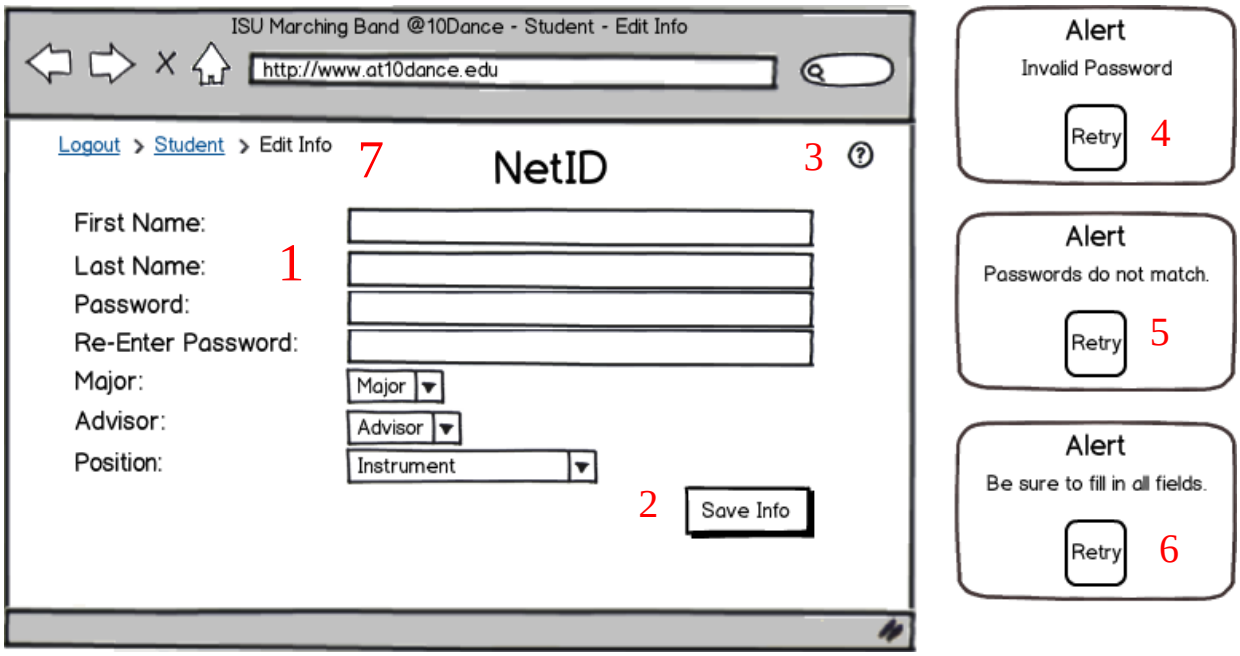
Purpose: To visualize the student's attendance data for the current season to date in a pie chart.

Loading Events: Clicking element 2 on (8_Student_Bar_Graph or 9_Student_Line_Graph)

Interface Elements:
1. Loads a bar graph visualization of the student's attendance data (9_Student_Line_Graph)
2. Clicking this tab does nothing on this screen, as the current view is already "pie chart."
3. Loads a line graph visualization of the student's attendance data (9_Student_Line_Graph)
4. Loads a list of all absences for this student (28_Student_View_Attendance_List)
5. Loads a list of all absence forms this student has submitted (29_Student_Absence_Form_List)
6. Returns to (7_Student_Page)
7. The bar graph visualization of the student's attendance record based on inputs 7 and 8.
8. Shows data within a selected date range.
9. Sorts the data by various parameters.

10. Sends user to a specialized help screen (27_Help) indicating how the user may view their attendance record and any submitted forms.
11. Page Trail. This allows the user to go back to the student page (7_Student_Page). Logout takes the user back to the home page (4_Main_Page).
12. Allows the user to select the date. Loads as today by default.

11_Student_Edit_Info
Sketch: Todd Wegter
Annotations: Curtis Ullerich



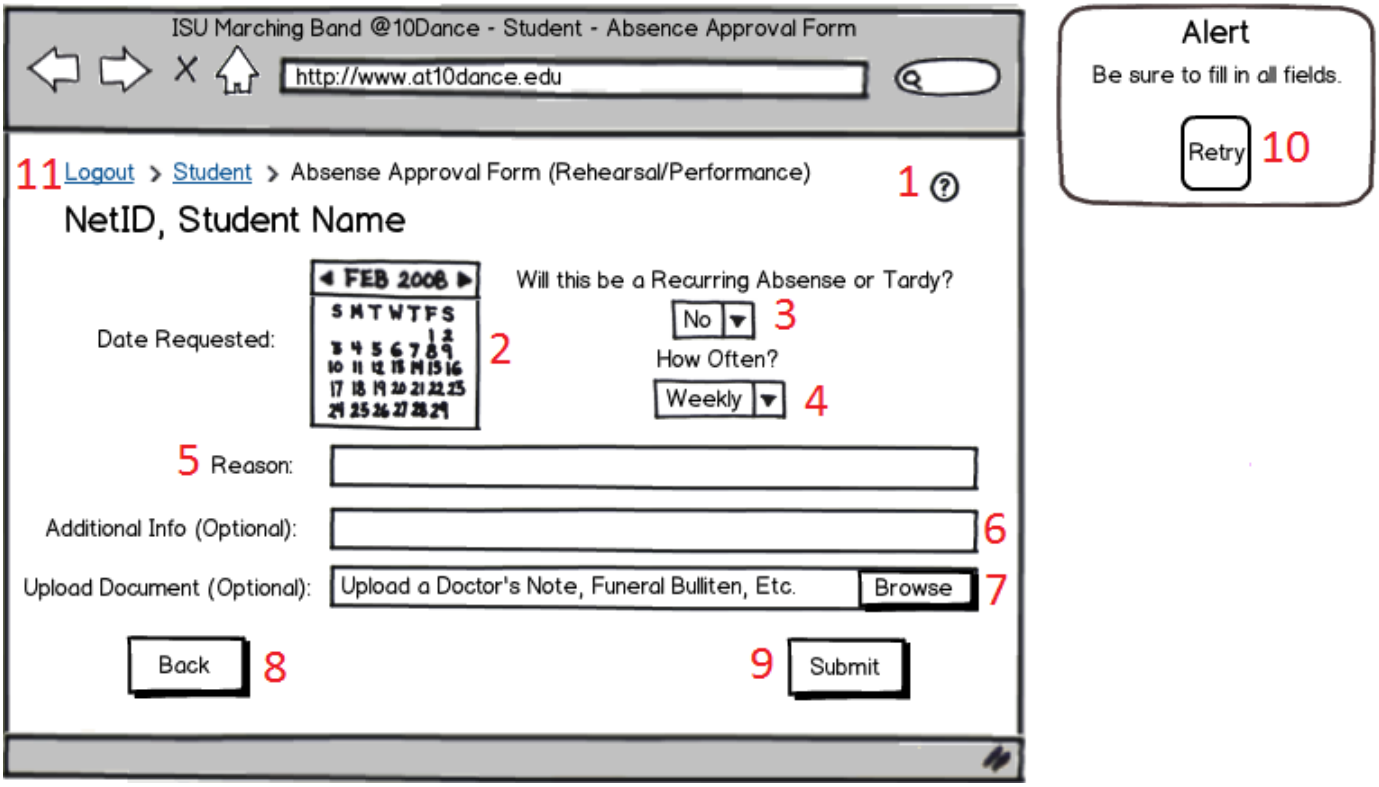
Purpose: Allows a student to edit their system information. Fields will be prepopulated with the student's current information. Note that the NetID is not an editable field.

Loading Events: Clicking element 3 on screen (7_Student_Page)

Interface Elements:
1. The fields that the student may edit.
2. Writes the information currently entered to the database.
3. Sends user to a specialized help screen (27_Help) indicating how the user may edit his or her information.
4. Alerts the user that the entered password does not comply with system security standards.
5. Alerts the user that the entered passwords do not match.
6. Alerts the user that a field has been left blank.

7. Page Trail. This allows the user to go back to the student page (7_Student_Page). Logout takes the user back to the home page (4_Main_Page).

12_Student_Absence_Approval_Form
Sketch: Todd Wegter, Yifei Zhu
Annotations: Brandon Maxwell



Purpose: This screen allows the student to request permission to repeatedly miss or arrive late at rehearsal or at the performance.

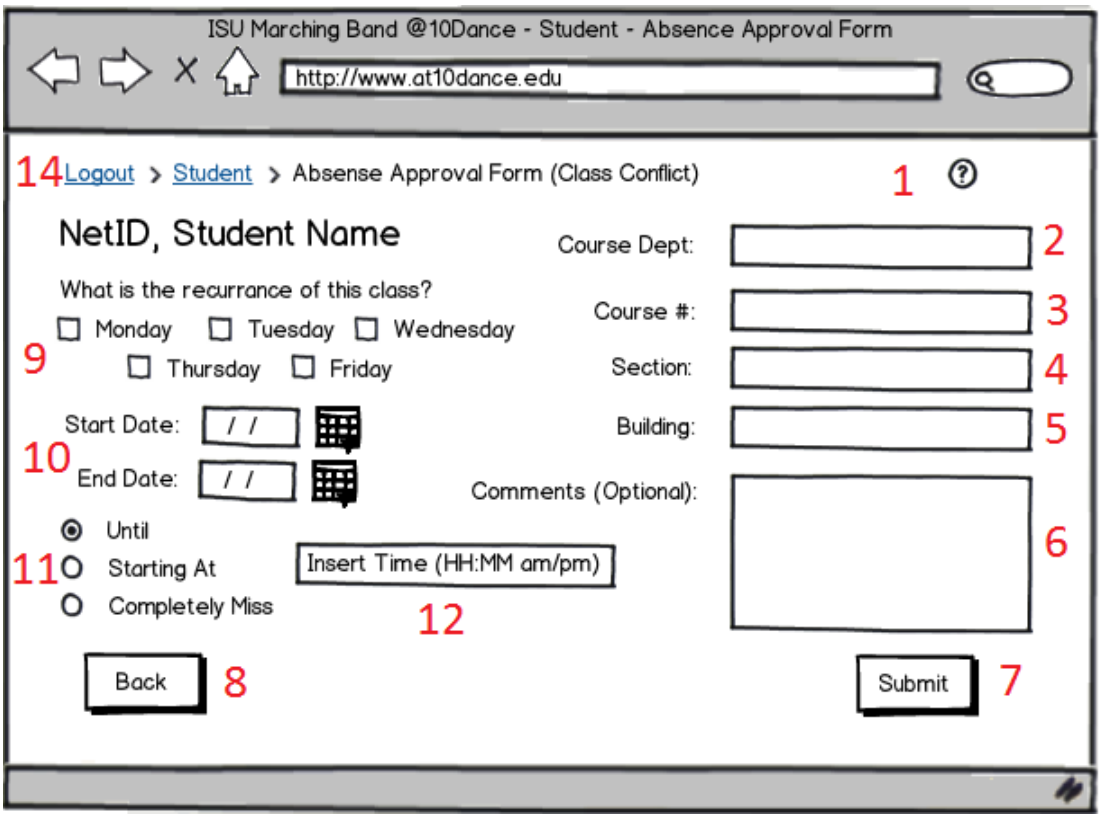
Loading Events: This window opens when the student checks either the rehearsal or performance checkbox and then clicks the next button on the 14_Student_Absence_Approval_Form_Type.

Interface Elements:

1. Help Button: when pressed this brings up the 27_Help screen.
2. Calendar: The student can select the date that he/she needs to miss by clicking on the day. To change months the student can click the arrows next to "Feb 2008."
3. This drop down box allows the student to tell if their absence is reoccurring.
4. If the student marks that their absence is reoccurring they can also select how often they will need to miss class, such as weekly or monthly.

5. In this text box the student can give the reason why they need to miss rehearsal or the performance.
6. This text box is provided for any kind of extra information the director would need about the request.
7. This box is provided for if the student needs to upload some kind of documentation to prove that their reason is valid.
8. A back button which will take the user to back to the 14_Student_Absence_Approval_Form_Type page.
9. A submit button, which will send the current request to the director. After the request has been sent the user will be taken back to the 14_Student_Absence_Approval_Form_Type.
10. The alert screen will pop up only if the student incorrectly filled out the Absence Approval Form. Pressing the retry button will take the student back to the 12_Student_Absence_Approval_Form.
11. Page Trail. This allows the user to go back to the Student Main Page (7_Student_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

13_Student_Class_Conflict_Form
Sketch: Todd Wegter, Yifei Zhu
Annotations: Brandon Maxwell



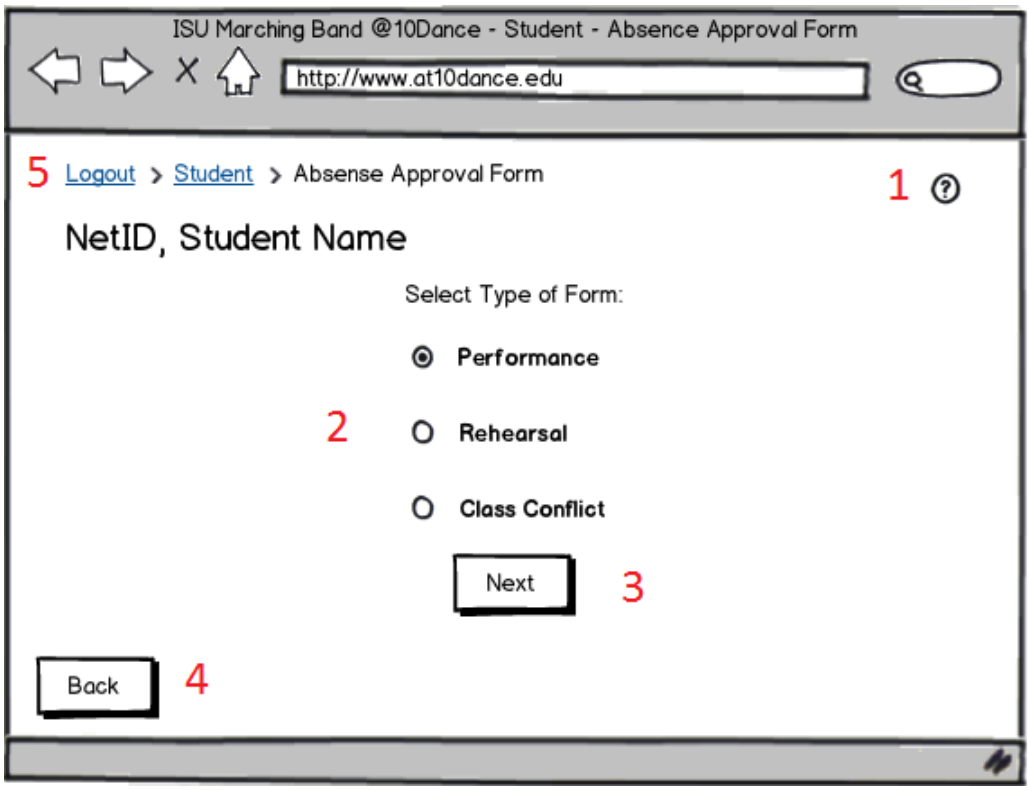
Purpose: This form is submitted once at the beginning of the of the season. Students are allowed to miss up to two rehearsals per week and they can fill out this form to get those absences approved.

Loading Events: This window opens when the student checks the class conflict checkbox and then clicks the next button on the 14_Student_Absence_Approval_Form_Type.

Interface Elements:

1. Help Button: when pressed this brings up the 27_Help screen.
2. A text box so the user can type the department of the course that conflicts with band.
3. A text box so the user can type the course number.
4. A text box so the user can type the section for the course.
5. A text box so the user can type what building they will be in
6. A text box so the user can add any extra comments that the directors might need.
7. A button that submits the current Absence Approval Form.
8. A back button which will take the user to back to the 14_Student_Absence_Approval_Form_Type page.
9. Several checkboxes where the user can show what day/days of the week they will be missing.
10. 2 text boxes where the user can type the starting and ending dates that they will have conflicts.
11. 3 checkboxes where the user can tell whether they will not arrive "until" a certain time, or have to leave "starting at" a certain time, or completely miss the rehearsal.
12. A textbox where the user can input the corresponding time depending on if they checked the until or starting at checkbox.
13. An alert screen that pops up if the user incorrectly fills out the Class Conflict Form.
14. Page Trail. This allows the user to go back to the Student Main Page (7_Student_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

14_Student_Absence_Approval_Form_Type
Sketch: Todd Wegter, Yifei Zhu
Annotations: Brandon Maxwell



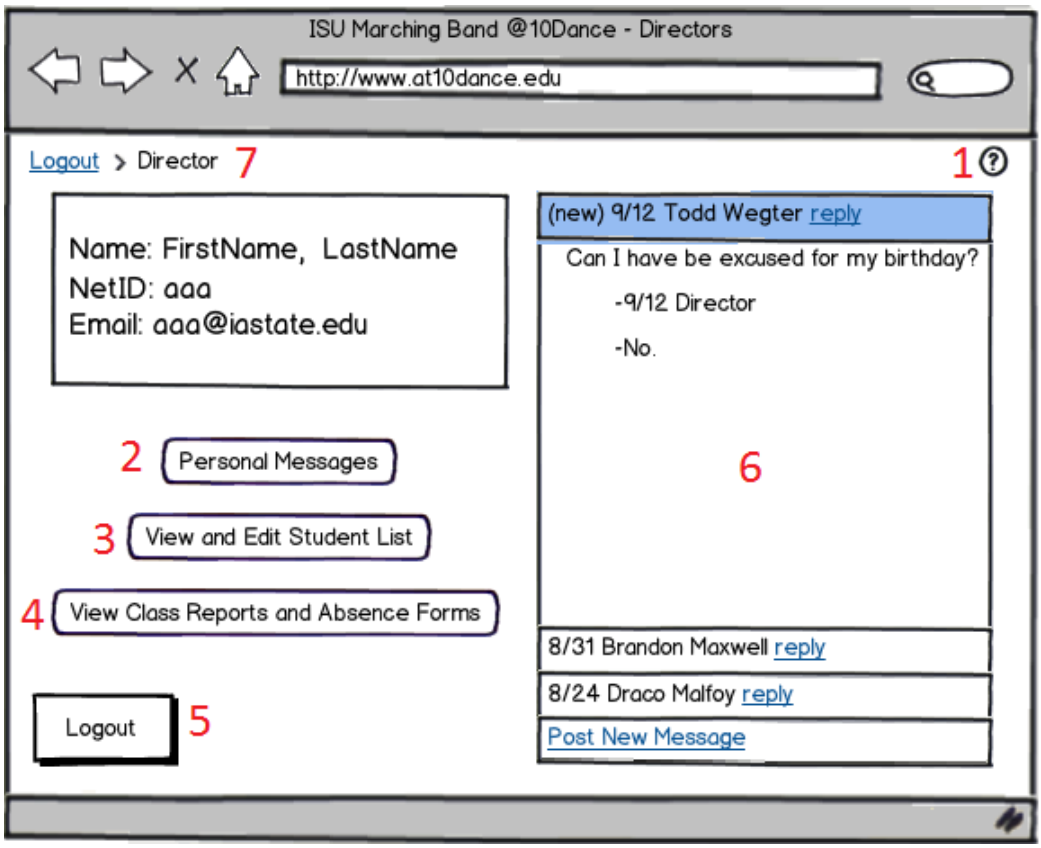
Purpose: This screen allows the user to choose which type of Absence Approval Form they need to submit, based on whether it a performance, rehearsal, or reoccurring class conflict.

Loading Events: The user is taken to this screen when they press the Absence Approval Forms button on the 7_Student_Page.

Interface Elements:

1. Help Button: when pressed this brings up the 27_Help screen.
2. 3 checkboxes that the user can use to tell the program which kind of form they will need to fill out.
3. A button the user will press once they have selected the type of Absence Approval Form they need to fill out.
4. A back button which will take the user to back to the 14_Student_Absence_Approval_Form_Type page.
5. Page Trail. This allows the user to go back to the Student Main Page (7_Student_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

15_Director_Page
Sketch: Todd Wegter,Yifei Zhu
Annotations: Brandon Maxwell

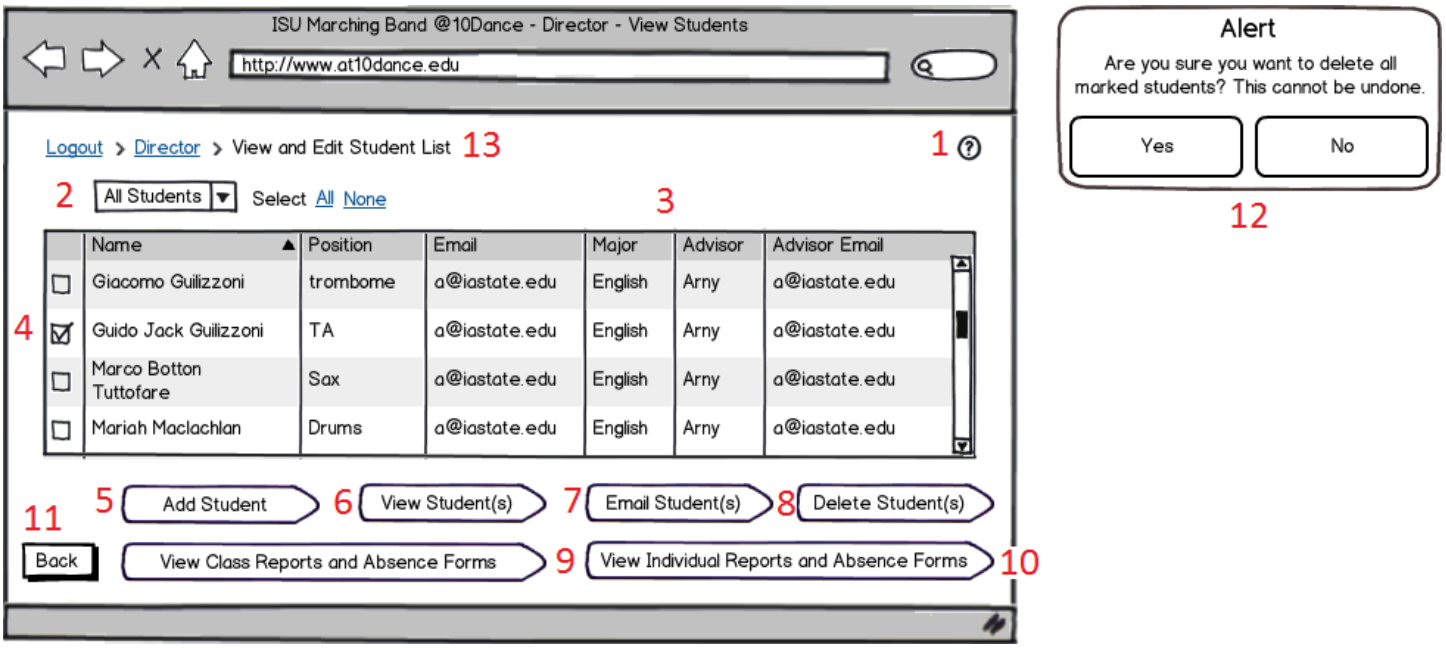


Purpose: This screen is the first page the director sees after logging into the system. It displays messages, their name and various buttons that allow the user to navigate to a different page.

Loading Events: Successful login (6_Login_Registration).

Interface Elements:
1. Help Button: when pressed this brings up the 27_Help screen.
2. A button that allows the user to access their current messages.
3. A button that when pressed will take the user to the 16_Director_Student_List page.
4. A button that when pressed will take the user to the 21_Director_View_Class_Report.
5. A back button which will take the user to back to the 14_Student_Absence_Approval_Form_Type page.
6. A screen that shows instant messages sent by students that the director can reply to immediately.
7. Page Trail. “Logout” logs the user out and returns them to the main screen (4_Main_Page).

16_Director_Student_List
Sketch: Todd Wegter, Yifei Zhu
Annotations: Brandon Maxwell



Purpose: This screen shows a grid layout of all the students in the class. It allows the user to perform different actions based on what buttons they press.

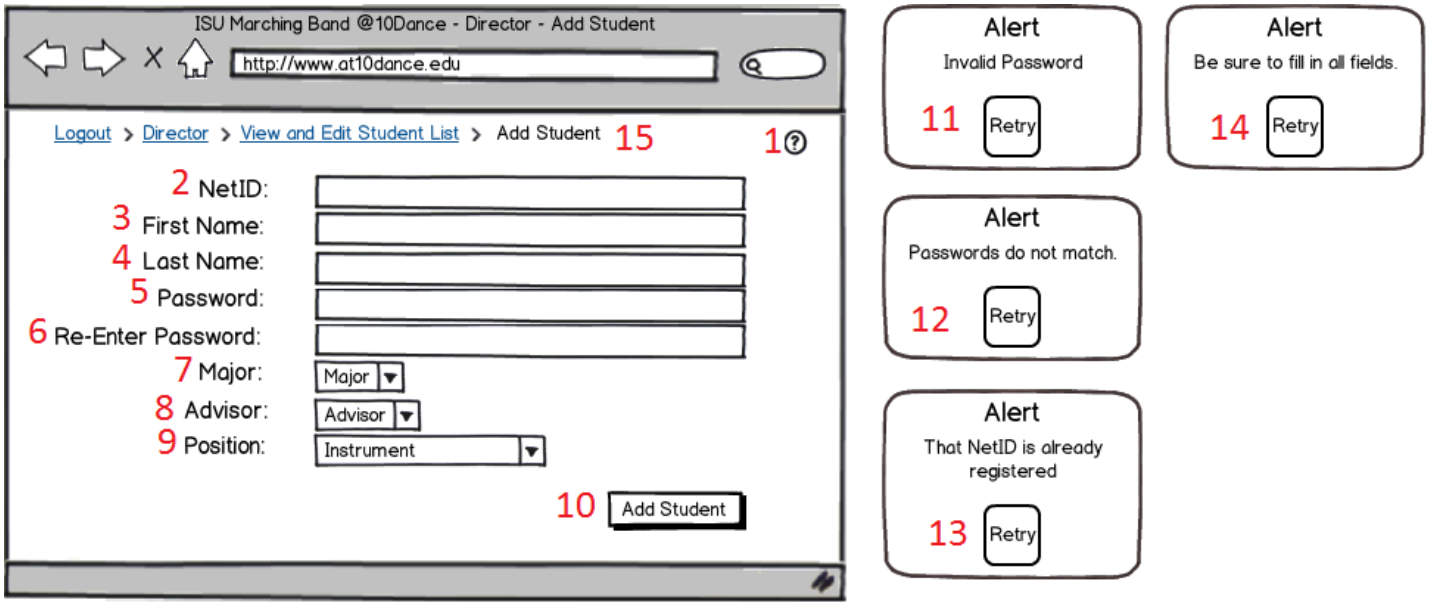
Loading Events: Pressing the "View and Edit Student List" button on the 15_Director_Page page.

Interface Elements:

1. Help Button: when pressed this brings up the 27_Help screen.
2. A drop down box that allows the user to select which students they want to view. They can also press the "all" or "none" buttons to selected all or none of the students.
3. A grid layout of all the students showing the name, position, email, major, advisor, and advisor email of all the students.
4. A check box that is used in conjunction with buttons 6, 7, or 8.
5. A button that allows the user to add a student. Takes them to 17_Director_Student_Add page.
6. A button that takes the user to the 18_Director_Student_View page.
7. A button that will open up the user's email account so they can send an email to the selected student.
8. A button that allows the user to delete a student/students.
9. A button that will take the user to the 23_Director_View_Class_Absence_Forms page.
10. A button that will take the user to the 20_Director_Student_View_Report page.
11. A back button which will take the user back to the 15_Director_Page page.

12. An alert that pops up if all the students in the list are selected and the user presses the delete button.
13. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

17_Director_Student_Add
Sketch: Todd Wegter
Annotations: Brandon Maxwell



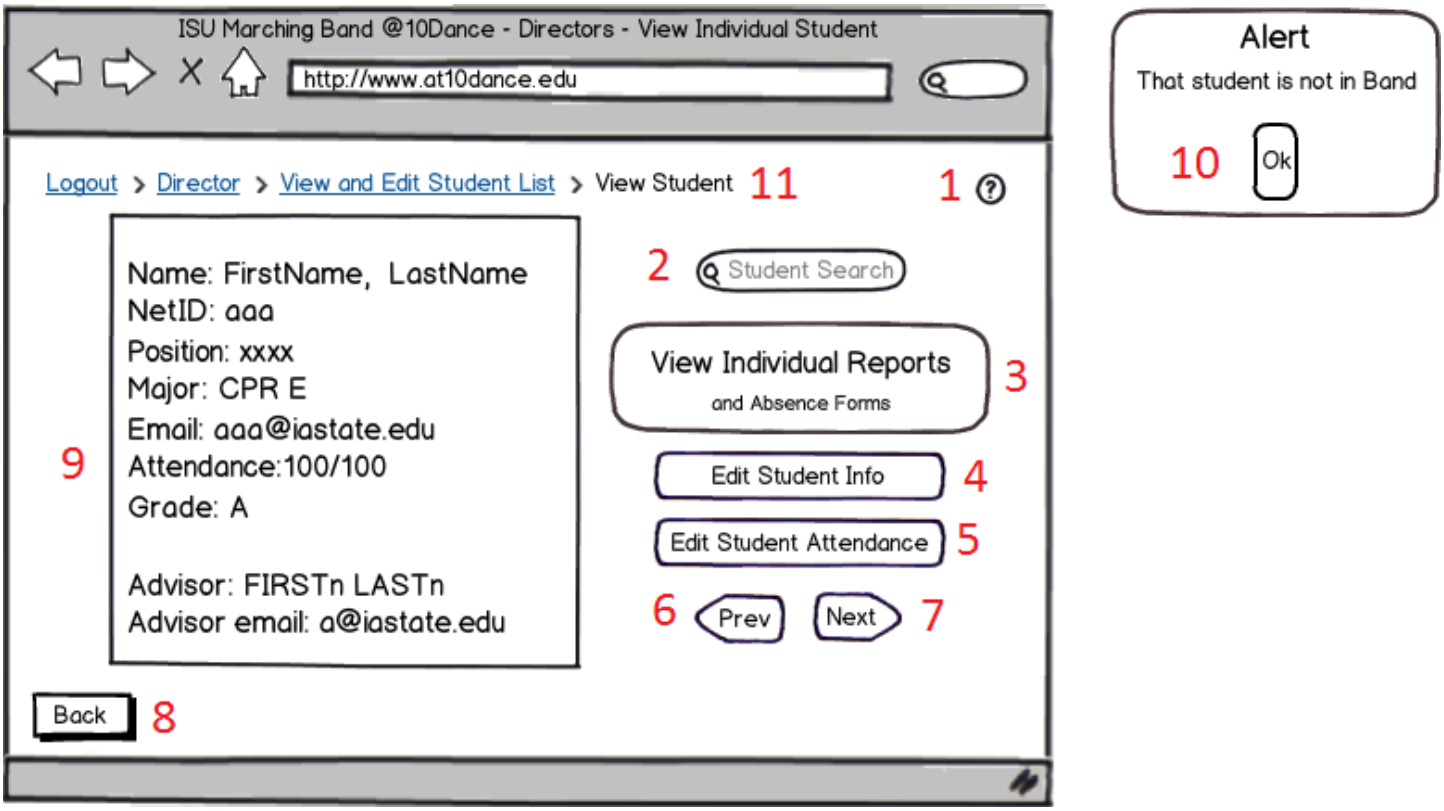
Purpose: This screen is used for when the director needs to add a student to the class.

Loading Events: Pressing the "Add Student" button on the 16_Director_Student_List page.

Interface Elements:
1. Help Button: when pressed this brings up the 27_Help screen.
2. A text box where the user can input the NetID of the student to be added.
3. A text box where the user can input the First Name of the student to be added.
4. A text box where the user can input the Last Name of the student to be added.
5. A text box where the user can input the Password of the student to be added.
6. A text box where the user can confirm the Password of the student to be added.
7. A dropdown box where the user can enter the Major of the student to be added.
8. A dropdown box where the user can enter the Advisor of the student to be added.
9. A dropdown box where the user can enter the Position of the student to be added.
10. A button to add the current student to the list of students in the class.
11. An alert that pops up if the user inputs an incorrectly formatted password.

12. An alert that pops up if the password and re-enter password fields don't match.
13. An alert that pops up if the user typed a NetID that is already taken.
14. An alert that pops up if the user doesn't fill out all of the fields.
15. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page) or the Student List Page (16_Director_Student_List). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

18_Director_Student_View
Sketch: Todd Wegter, Yifei Zhu
Annotations: Brandon Maxwell



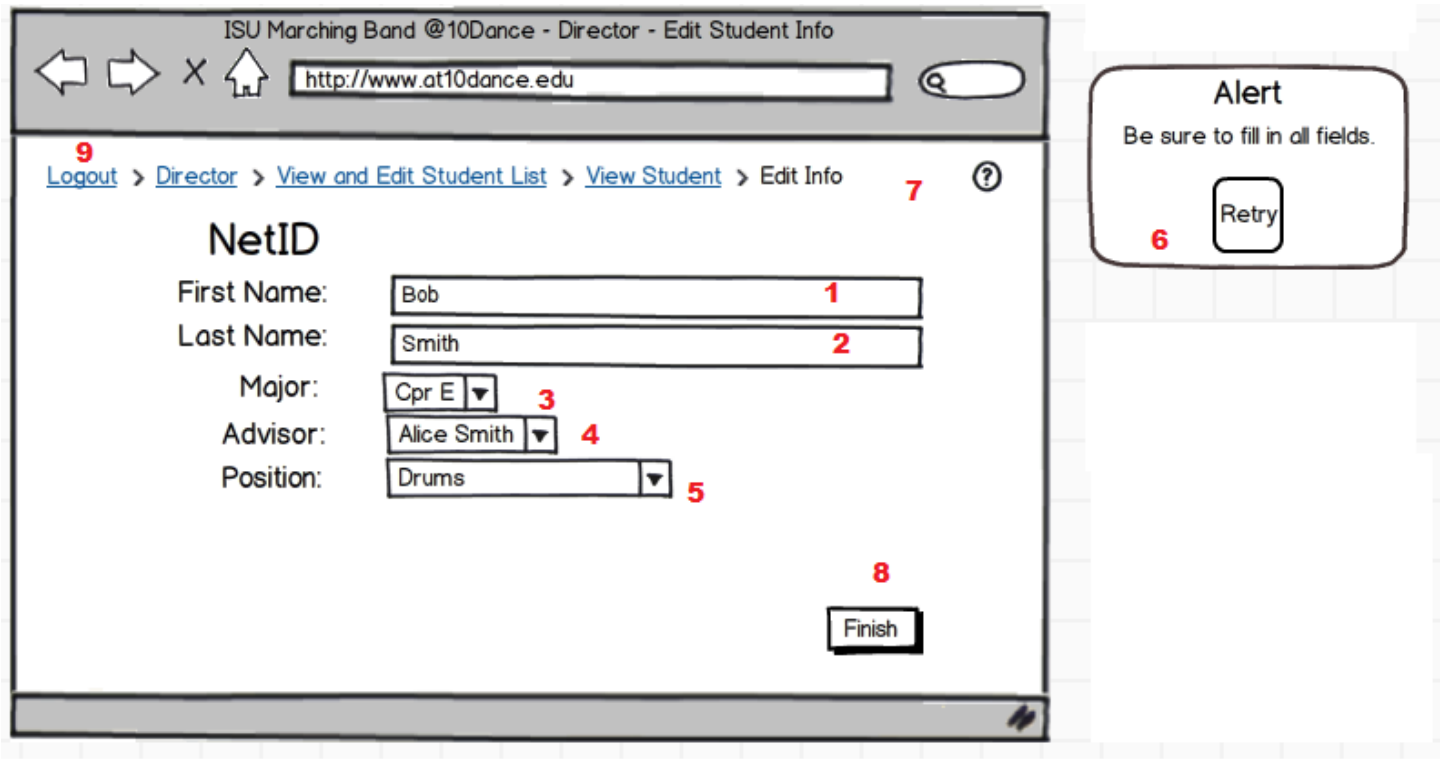
Purpose: This screen is used for when the director wants to view the current information about a student.

Loading Events: Pressing the "View Student" button on the 16_Director_Student_List page.

Interface Elements:
1. Help Button: when pressed this brings up the 27_Help screen.
2. A text box that allows the user to search for a specific student.
3. A button that when pressed will take the user to the 20_Director_Student_View_Report page.

4. A button that when pressed will take the user to the19_Director_Student_Edit_Info page.
5. A button that when pressed will allow user to edit student record.
6. A button that allows the user to go from this student to the previous student in the list.
7. A button that allows the user to go from this student to the next student in the list.
8. A back button that will take the user to the 16_Director_Student_List page.
9. A box that displays information about the student the user is currently viewing.
10. An alert that pops up if the user searches for a student that is not in the list.
11. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page) or the Student List Page (16_Director_Student_List). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

19_Director_Student_Edit_Info
Sketch: Todd Wegter
Annotations: Yifei Zhu



Purpose: Allow director to modify student's information

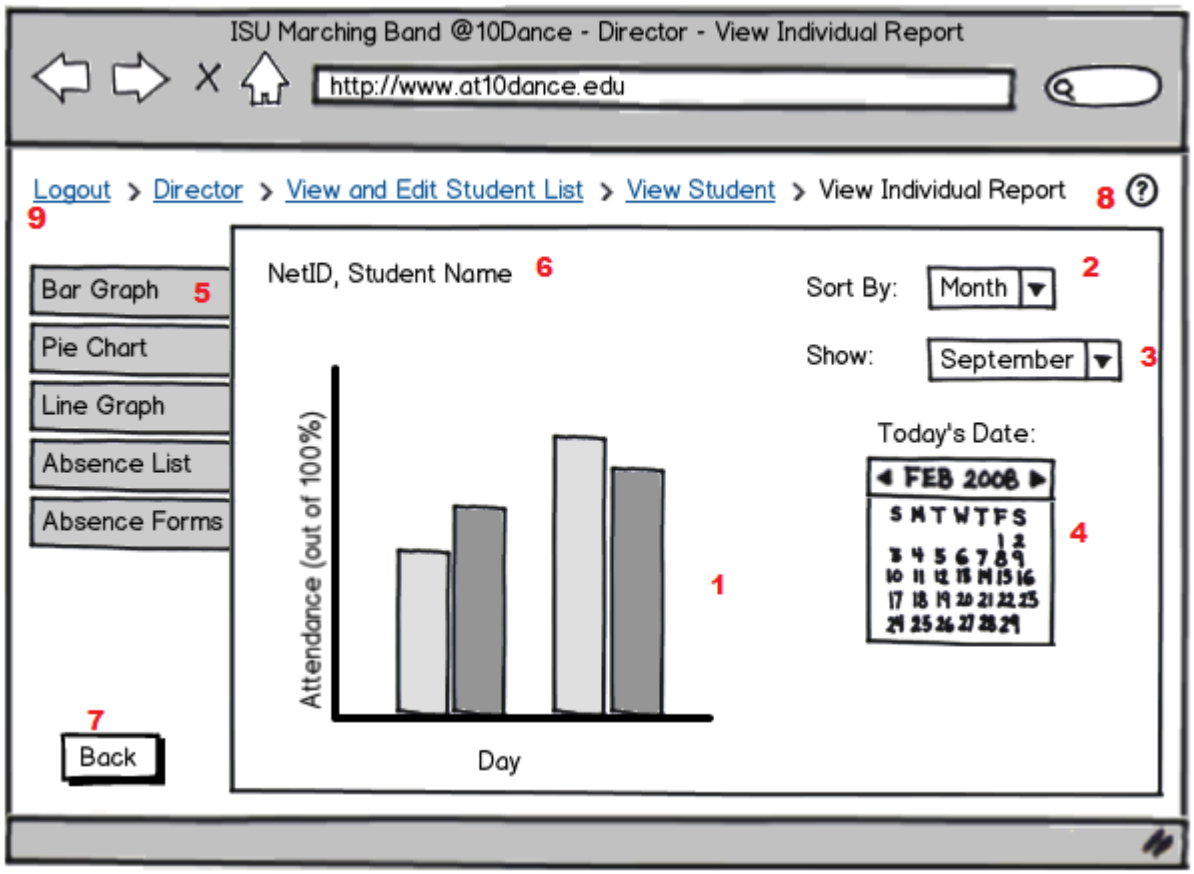
Loading Events: User presses "Edit Student Info" button on View Student Window (Screen 18_Director_Student_View)

Interface Elements:

1. The text field allows director to manually edit student's first name

2. The text field allows director to manually edit student's last name
3. The text field allows director to edit student's major
4. The text field allows director to edit student's advisor
5. The text field allows director to edit student's position
6. The alert message invalid input
7. If the director click the question mark, then a help window appear (Screen 27_Help).
8. The "Finish" button returns to the View Student web page(Screen 18_Director_Student_View)
9. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page) or the Student List Page (16_Director_Student_List) or View Student (18_Director_Student_View). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

20_Director_Student_View_Report
Sketch: Todd Wegter, Yifei Zhu
Annotations: Yifei Zhu

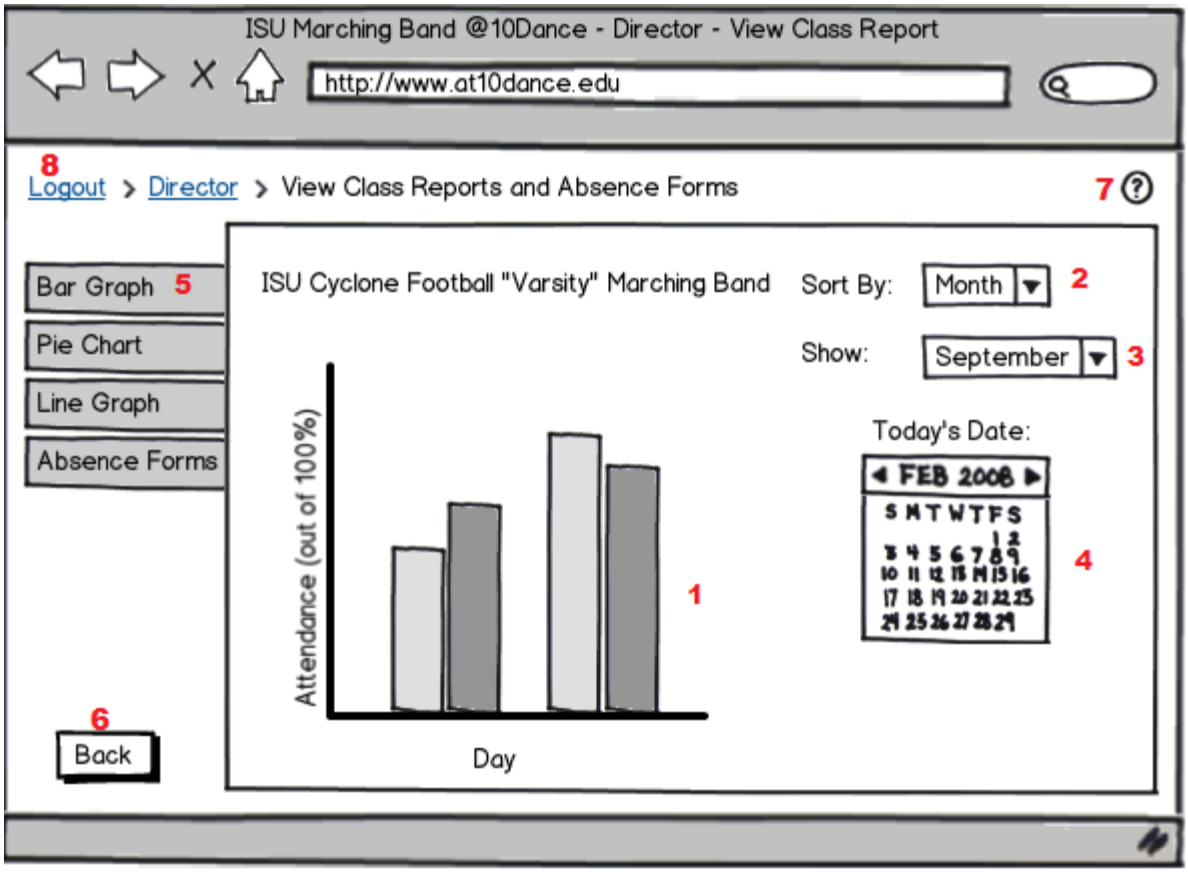


Purpose: Director can view student's attendance in bar graph
Loading Events: User presses "View Individual Reports and Absence Forms" button on View Student Window page(Screen 18_Director_Student_View)

Interface Elements:

1. The bar graph shows a student's attendance
2. A drop down menu for selecting what to sort by
3. A drop down menu for selecting the timeline
4. A calendar to display the date
5. The tabs shows different types of graphs and forms
6. The title shows the student's NetID and name
7. The "Back" button returns back to the student view (Screen 18_Director_Student_View)
8. If the director click the question mark, then a help window appear (Screen 27_Help).
9. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page) or the Student List Page (16_Director_Student_List) or View Student (18_Director_Student_View). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

21_Director_View_Class_Report
Sketch: Todd Wegter, Yifei Zhu
Annotations: Yifei Zhu



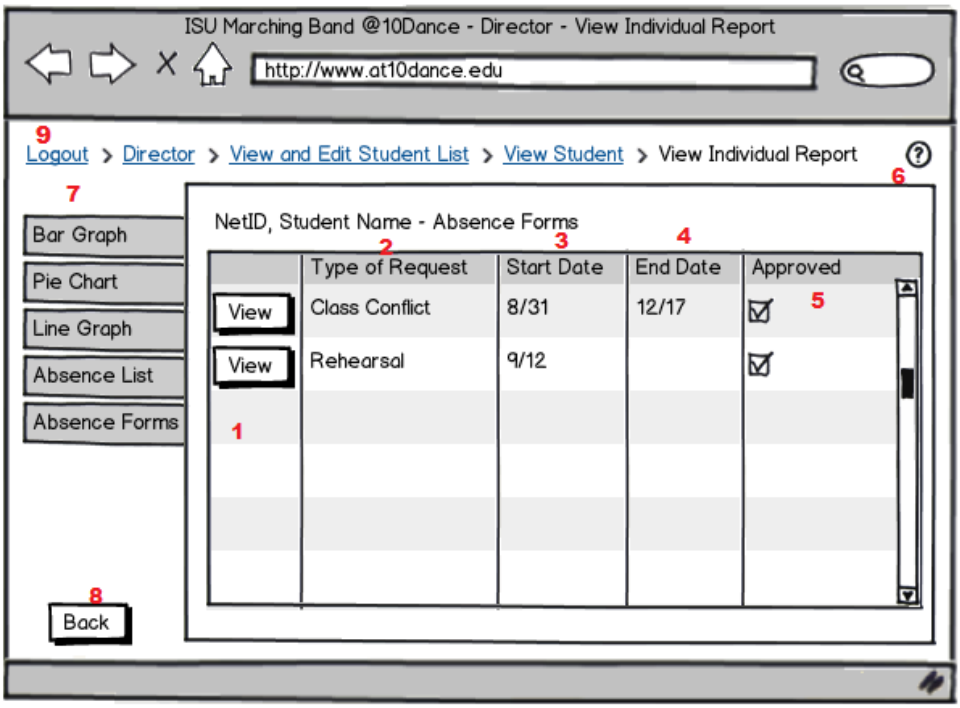
Purpose: Director can view student's attendance in bar graph

Loading Events: User presses "View Individual Reports and Absence Forms" button on View Student Window page(Screen 18_Director_Student_View)

Interface Elements:

1. The bar graph shows a student's attendance
2. A drop down menu for selecting what to sort by
3. A drop down menu for selecting the timeline
4. A calendar to display the date
5. The tabs shows different types of graphs and forms
6. The "Back" button returns back to the student view (Screen 18_Director_Student_View)
7. If the director click the question mark, then a help window appear (Screen 27_Help).
8. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

22_Director_Student_View_Absence_Form_List
Sketch: Todd Wegter, Yifei Zhu
Annotations: Yifei Zhu



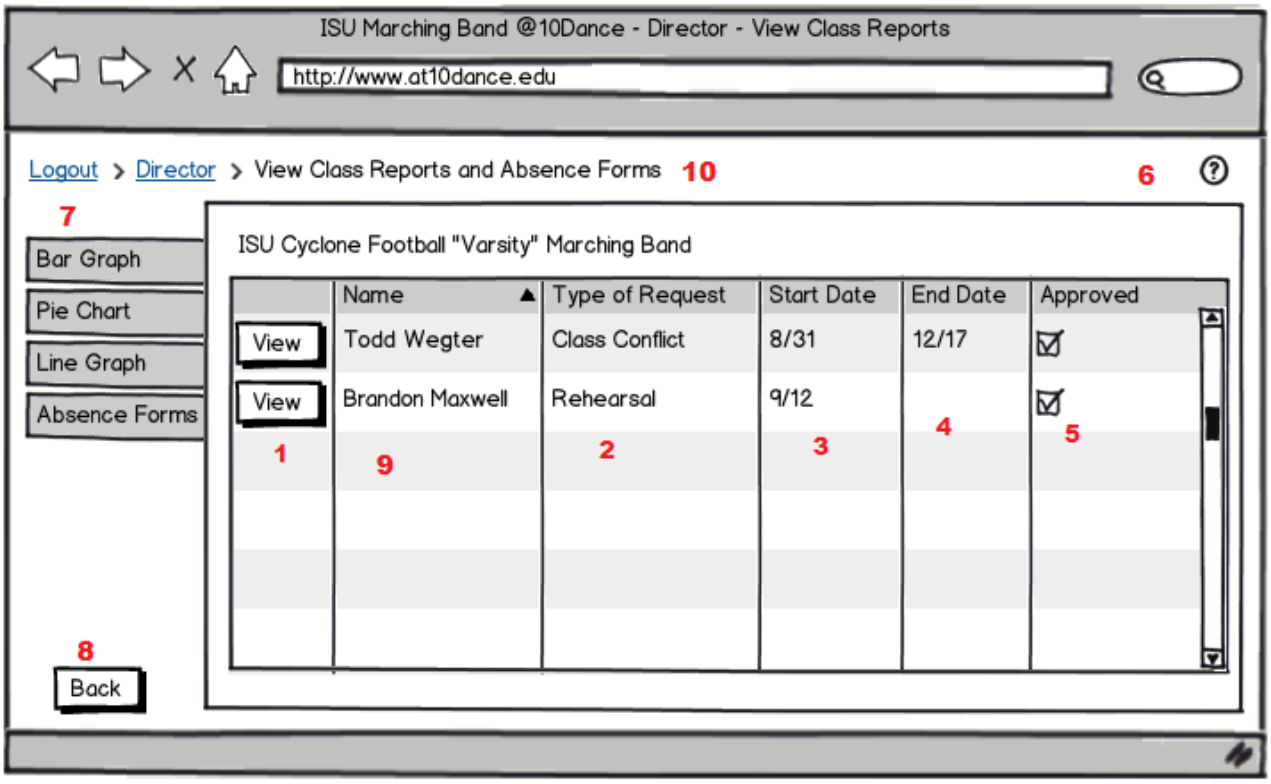
Purpose: Allow director to view individual student's absence forms

Loading Events: User presses "View Individual Reports and Absence Forms" button on View Student Window(Screen 18_Director_Student_View) or "View Individual Report and Absence forms on (screen 16_Director_Student_List).

Interface Elements:

1. If the director click the "View" button, the specific form will popup (screen 31_Director_Student_View_Absence) or (screen 24_Director_Student_View_Conflict)
2. This column displays the types of requests.
3. This column displays the request start date
4. This column displays the request end date
5. The checklist shows whether the requests have been approved.
6. If the director clicks the question mark, then a help window appears (Screen 27_Help).
7. The tabs show different types of graphs and forms.
8. The "Back" button returns to the View Student webpage(Screen 18_Director_Student_View)
9. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page) or the Student List Page (16_Director_Student_List) or View Student (18_Director_Student_View). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

23_Director_View_Class_Absence_Forms
Sketch: Todd Wegter, Yifei Zhu
Annotations: Yifei Zhu



Purpose: Allow director to view individual student's absence forms

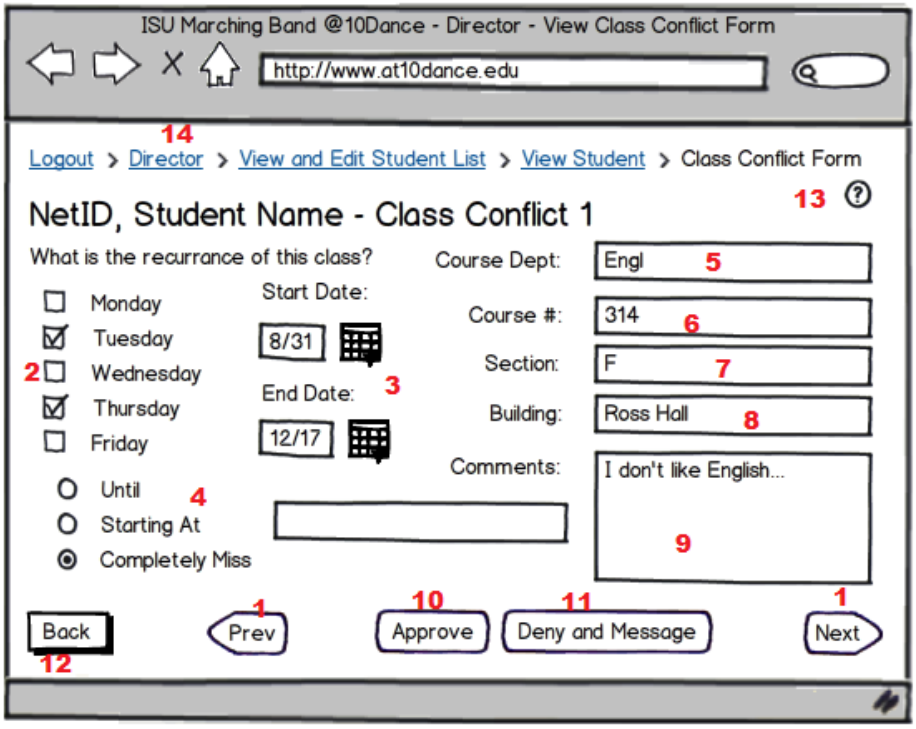
Loading Events: User presses "View Class Reports and Absence Forms" of the Director Student List (screen
16_Director_Student_List).

Interface Elements:

1. If the director click the "View" button, the specific form will popup student conflict form (screen 4_Director_Student_View_Conflict) or student absence form(screen 31_Director_Student_View_Absence_Form)
2. This column displays the types of requests.
3. This column displays the request start date
4. This column displays the request end date
5. The checklist shows whether the requests have been approved.
6. If the director clicks the question mark, then a help window appears (Screen 27_Help).
7. The tabs show different types of graphs and forms.
8. The "Back" button returns to the View Student webpage(Screen 18_Director_Student_View)
9. The name list shows the students who have been absent from band practice
10. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page).

"Logout" logs the user out and returns them to the main screen (4_Main_Page).

24_Director_Student_View_Conflict
Sketch: Todd Wegter, Yifei Zhu
Annotations: Yifei Zhu



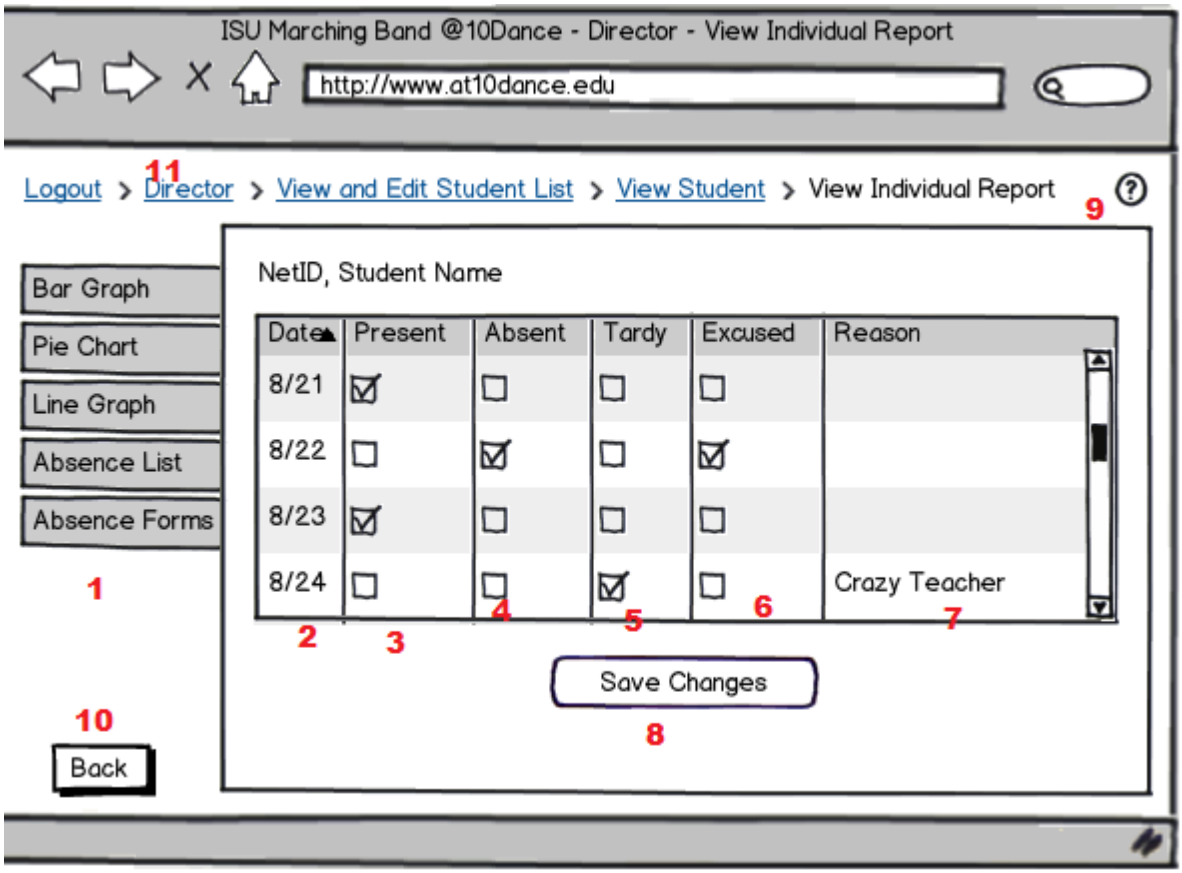
Purpose: Allow director to approve/disprove students' class conflict forms

Loading Events: User presses "View Individual Reports and Absence Forms " button on View Student Window(Screen 18_Director_Student_View)

Interface Elements:
1. Go to the previous or next form.
2. The checkmark shows the dates a student has a class conflict.
3. The Start Date and End Date the student has classes conflict
4. Shows what portion of class will be missed. (radio buttons and text box)
5. The text field shows the conflicts course department
6. The text field shows the conflicts course name
7. The text field shows the conflicts course section
8. The text field shows the conflicts course address
9. The text field shows comments from the student for the conflict
10. The Director clicks on the "Approve" button, if he agrees with the class conflict
11. The Director clicks on the "Deny and Message" button, if he disagrees with the class conflict
12. The "Back" button returns back to the student view page(Screen 18_Director_Student_View)
13. If the director clicks the question mark, then a help window appears (Screen 27_Help).

14. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page) or the Student List Page (16_Director_Student_List) or View Student (18_Director_Student_View). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

25_Director_Student_Absence_List
Sketch: Todd Wegter
Annotations: Yifei Zhu



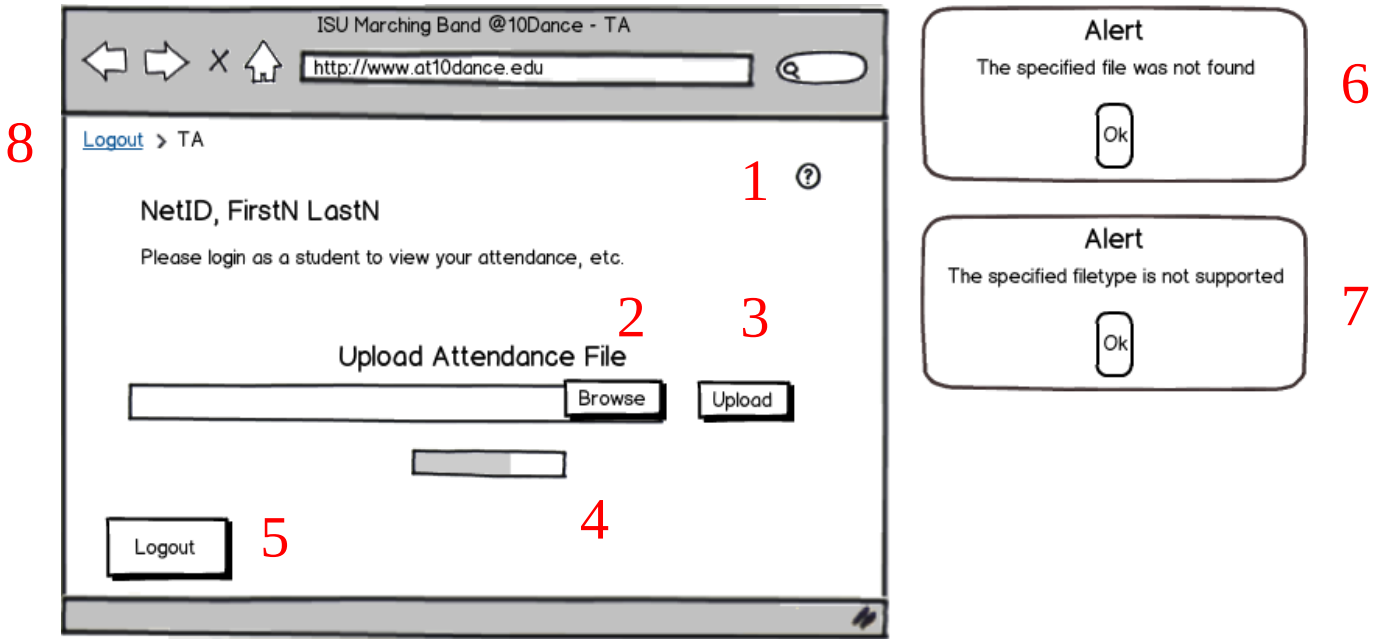
Purpose: Allows the director to view a student's individual report

Loading Events: User presses "View Individual Reports and Absence Forms " button on View Student Window(Screen 18_Director_Student_View)

Interface Elements:
1. The tabs show different types of graphs and forms.
2. The dates the students have class.
3. The checkmark indicates that the student came to class
4. The checkmark indicates that the student was absent from class

5. The checkmark indicates that the student was tardy from class
6. The checkmark indicates that the student was excused from class
7. The text field shows the reason the student could not get to class on time
8. Used to save all of the changes to the attendance.
9. If the director clicks the question mark, then a help window appears (Screen 27_Help).
10. The "Back" button returns to the View Student webpage(Screen 18_Director_Student_View)
11. Page Trail. This allows the user to go back to the Director Main page (15_Director_Page) or the Student List Page (16_Director_Student_List) or View Student (18_Director_Student_View). "Logout" logs the user out and returns them to the main screen (4_Main_Page).

26_TA_Page
Sketch: Todd Wegter
Annotations: Todd Wegter



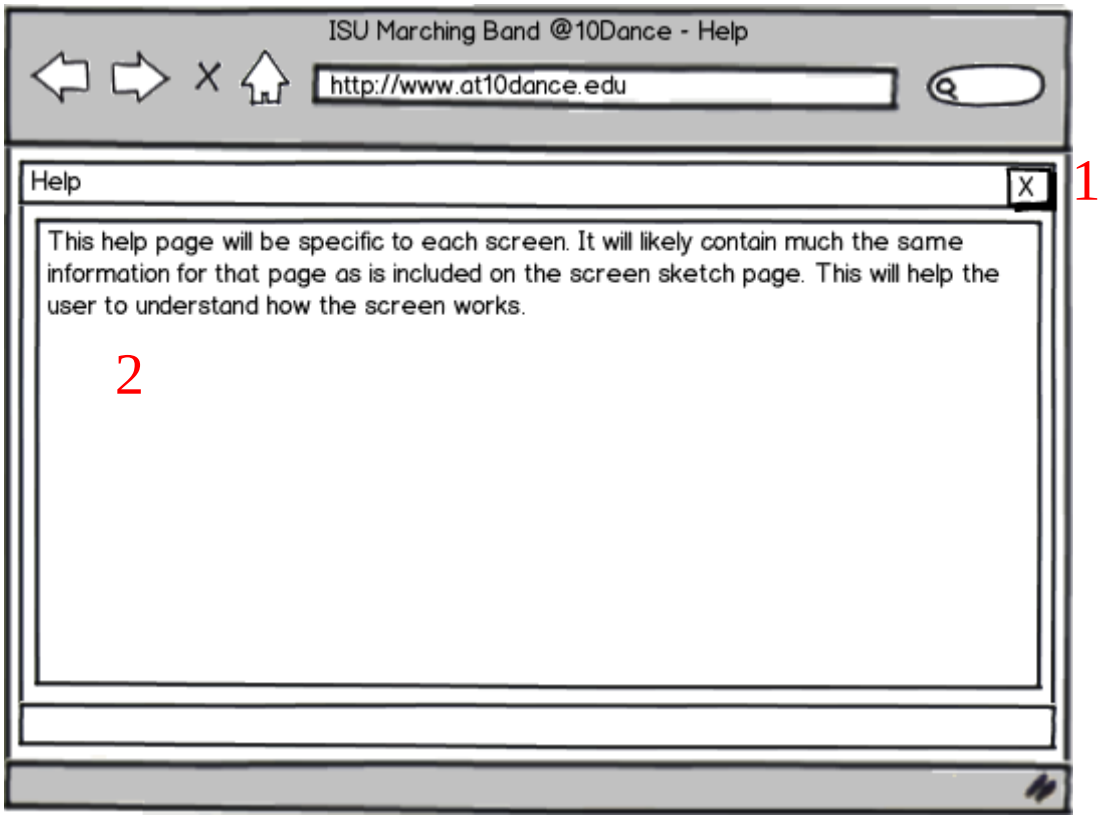
Purpose: This is the TA home page. It allows TAs to upload attendance data exported from the html5 app. TAs can view their attendance, etc. by logging in as a student.

Loading Events: User logs in to the online application as a TA.

Interface Elements:
1. Help Button. Pulls up a help page (27_Help) detailing the items on this page and how to use them.
2. File to Upload Selector. Allow user to either type in the filepath of a file to upload or select it in windows explorer by clicking the "Browse" button.

3. Upload Button. Uploads the selected attendance data file
4. Progress Bar. Shows the progress of the attendance file upload.
5. Logout Button. Allows user to logout, returning them to the main screen (4_Main_Page).
6. File Not Found Alert. The file specified to upload in (2) could not be found when (3) is clicked.
7. File Type Alert. The type of the file specified to upload in (2) when (3) is clicked is not supported.
8. Page Trail. This allows the user to go back to any page higher up in the page hierarchy. "Logout" logs the user out and returns them to the main screen (4_Main_Page).

27_Help
Sketch: Todd Wegter
Annotations: Todd Wegter



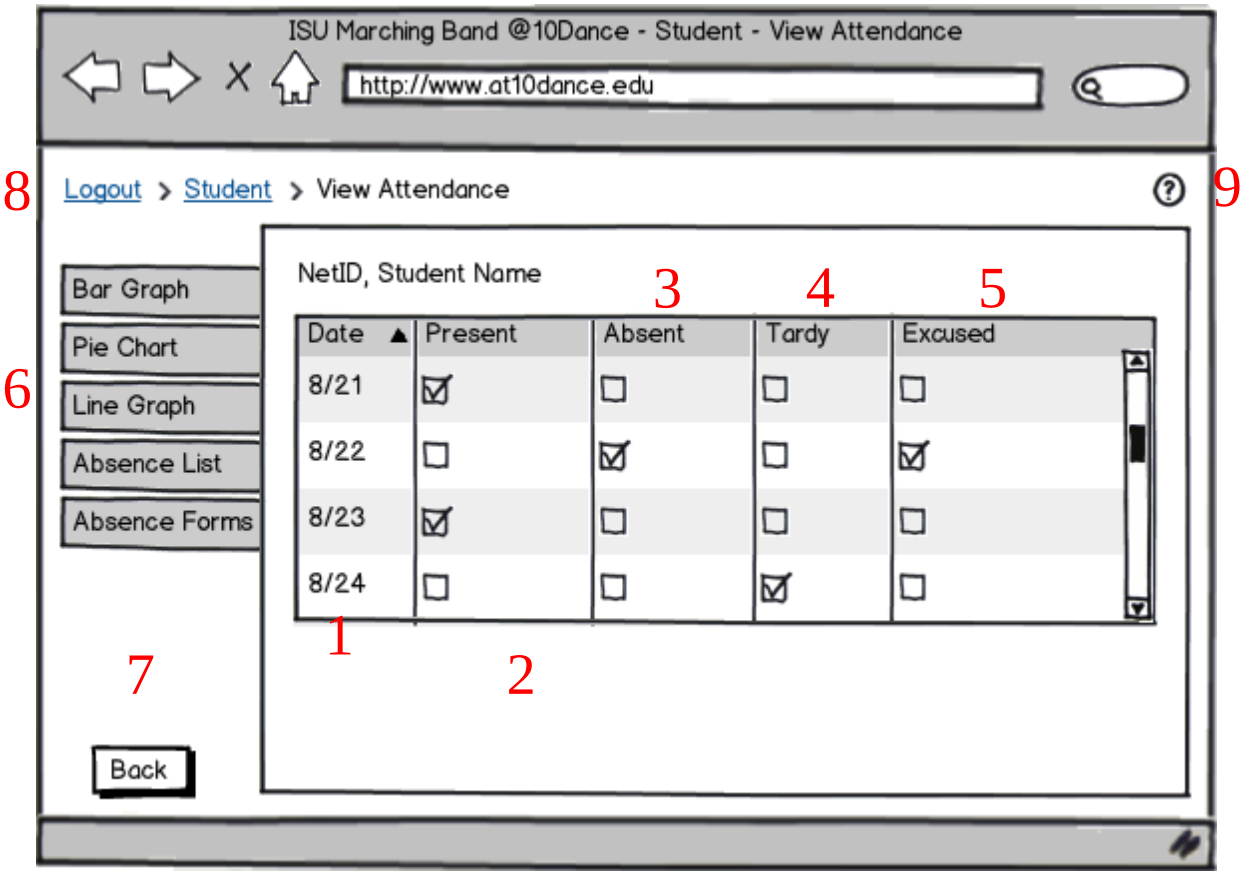
Purpose: This screen pops up over other screens when the Help Button is pushed. It is custom tailored to each screen.

Loading Events: This screen pops up over other screens when the Help Button is pushed.

Interface Elements:

1. Exit Button. Closes the help window (2), revealing the page it was loaded over.
2. Help Window. Gives a detailed account of how to use the page in which the Help Button was clicked.

28_Student_View_Attendance_List
Sketch: Todd Wegter
Annotations: Todd Wegter



Purpose: This is the page where students can view the list of their attendances by date.

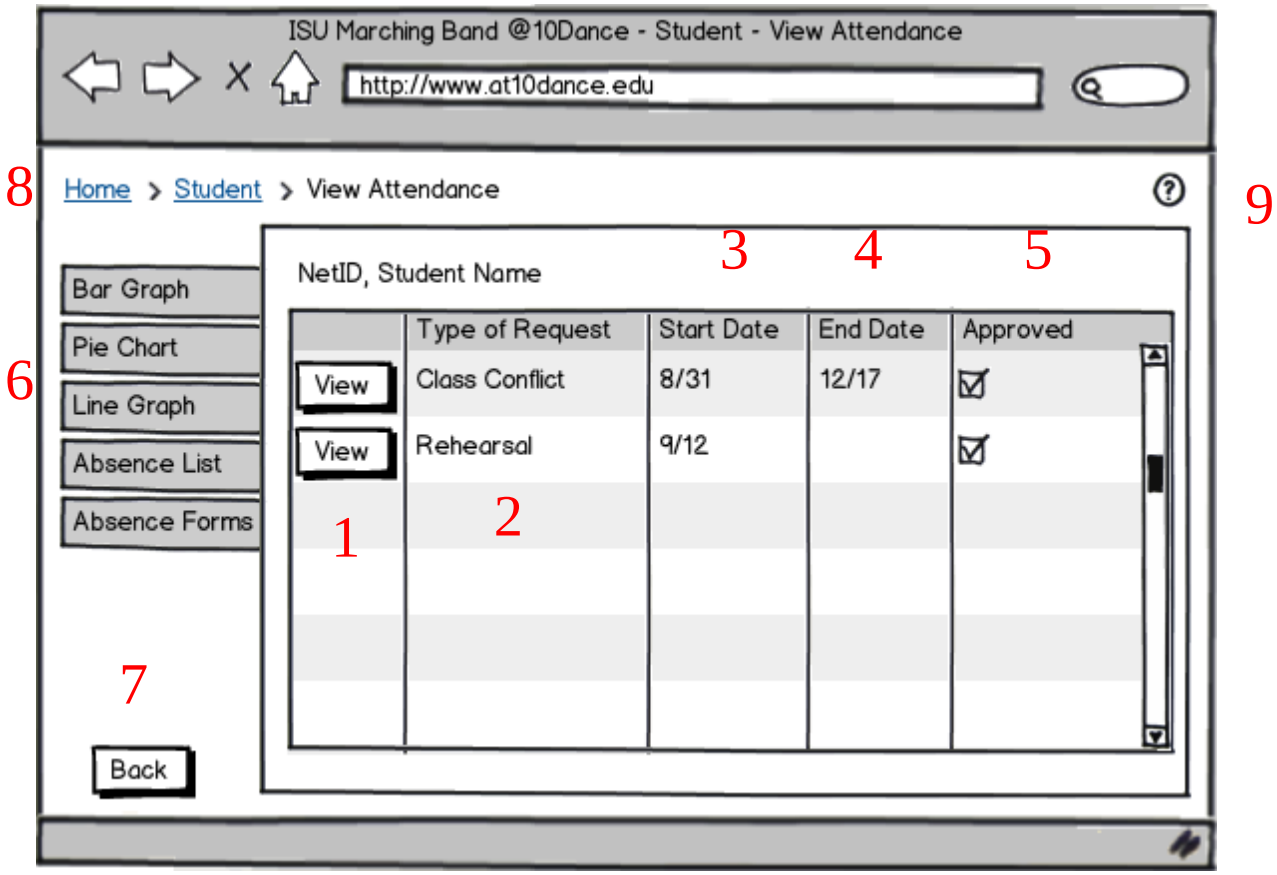
Loading Events: This page is loaded by clicking the "View Attendance" button on the Student Main Page (7_Student_Page). It can also be switched to from any other attendance page using the Attendance Page Tabs (6).

Interface Elements:

1. Date Column. This shows the date for each attendance record.
2. Present Column. Shows the days the user was present.

3. Absent Column. Shows the days the user was absent.
4. Tardy Column. Shows the days the user was tardy.
5. Excused Column. Shows the absences that were excused.
6. Attendance Page Tabs. Allows the user to switch between (8_Student_Bar_Graph) (9_Student_Line_Graph) (10_Student_Pie_Chart), and (29_Student_Absence_Form_List).
7. Back Button. Takes the user back to the Student Main Page (7_Student_Page).
8. Page Trail. This allows the user to go back to the Student Main Page (7_Student_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).
9. Help Button. Pulls up a help page (27_Help) detailing the items on this page and how to use them.

29_Student_Absence_Form_List
Sketch: Todd Wegter, Yifei Zhu
Annotations: Todd Wegter



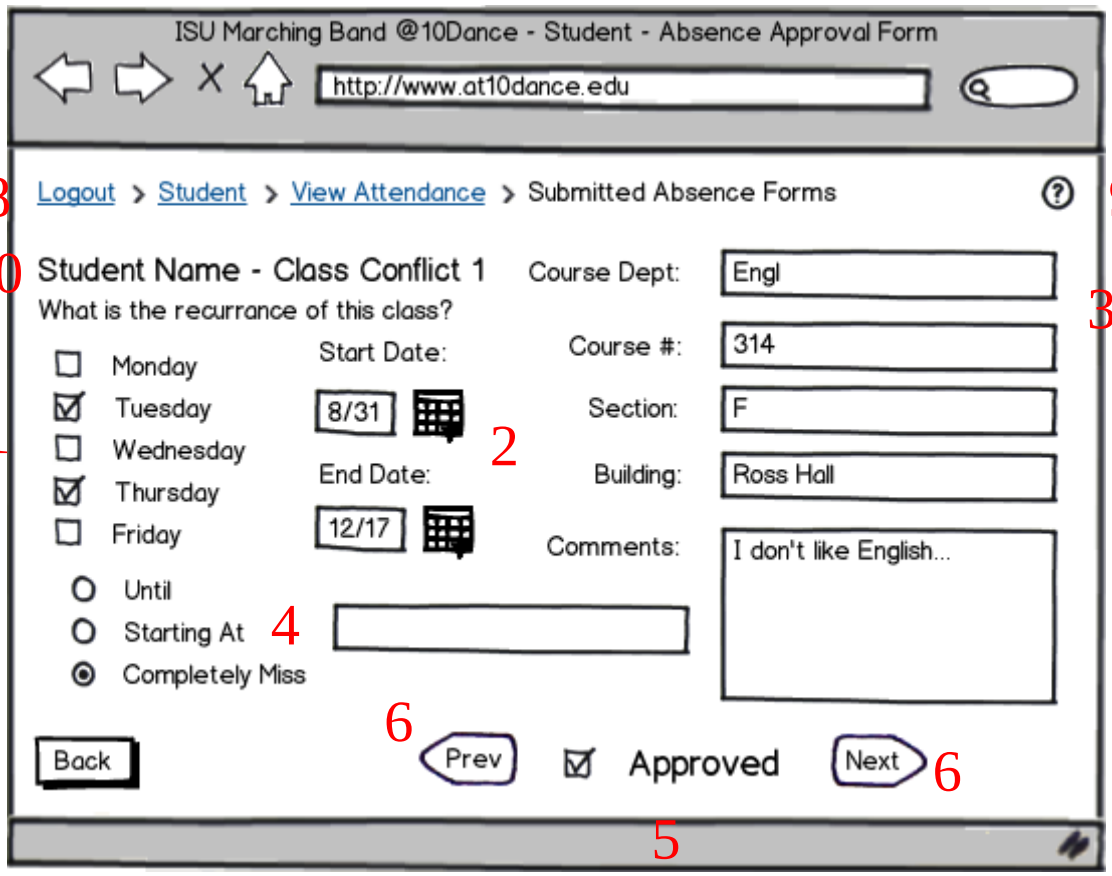
Purpose: This is the student view of absence forms he or she has submitted.

Loading Events: This page can also be switched to from any other attendance page using the Attendance Page Tabs (6).

Interface Elements:

1. View Buttons. Allows the user to view the absence form in the specified row.
2. Type of Request Column. Shows the type of the absence form.
3. Start Date Column. Shows the start date for repeating absence forms and the only day for single day occurrences.
4. End Date Column. Shows the end date for repeating absence forms.
5. Approved Column. Shows the absence requests that have been approved.
6. Attendance Page Tabs. Allows the user to switch between (8_Student_Bar_Graph) (9_Student_Line_Graph) (10_Student_Pie_Chart), and (28_Student_View_Attendance_List).
7. Back Button. Takes the user back to the Student Main Page (7_Student_Page).
8. Page Trail. This allows the user to go back to the Student Main Page (7_Student_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).
9. Help Button. Pulls up a help page (27_Help) detailing the items on this page and how to use them.

30_Student_View_Absence_Approval_Forms
Sketch: Todd Wegter, Yifei Zhu
Annotations: Todd Wegter



Purpose: This page allows the student to view each individual submitted absence form (the shown page is specific to Class Conflict forms).

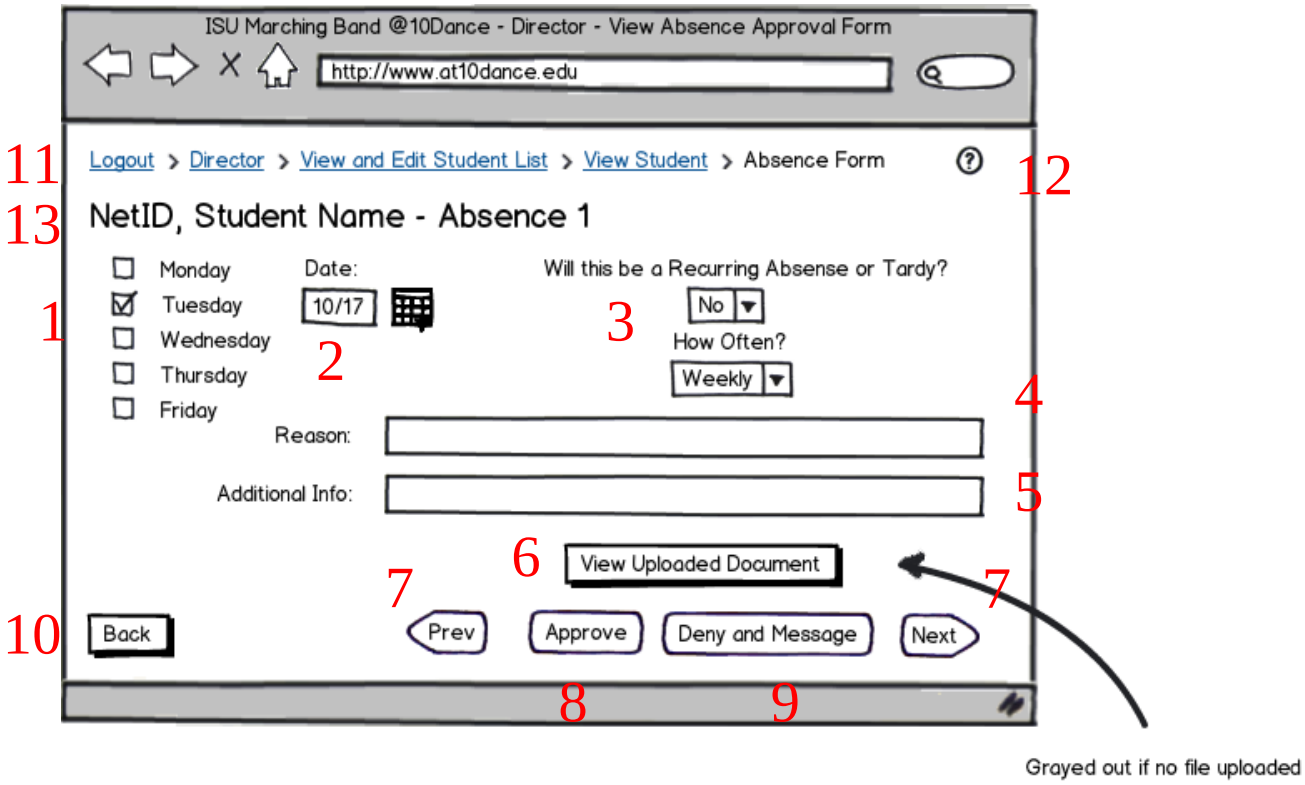
Loading Events: Clicking the "View" Button for any absence form in (29_Student_Absence_Form_List).

Interface Elements:

1. Effective Days. Shows the days of the week that this form applies to.
2. Effective Dates. Shows the dates that this form is valid for.
3. Class Information. Shows the information for the class conflict.
4. Missing Portion. Shows what portion of class will be missed. (radio buttons and text box)
5. Approved Checkbox. A checkmark shows that the form has been approved.
6. Prev/Next Buttons. Allows the user to view the next or previous submitted absence form.
7. Back Button. Takes the user back to the List of Student Absence Forms page (29_Student_Absence_Form_List).

8. Page Trail. This allows the user to go back to the Student Main Page (7_Student_Page) or the List of Student Absence Forms page (29_Student_Absence_Form_List). "Logout" logs the user out and returns them to the main screen (4_Main_Page).
9. Help Button. Pulls up a help page (27_Help) detailing the items on this page and how to use them.
10. Absence Form Label. Shows the User's name and the number and type of the absence form he or she is viewing.

31_Director_Student_View_Absence_Form
Sketch: Todd Wegter
Annotations: Todd Wegter



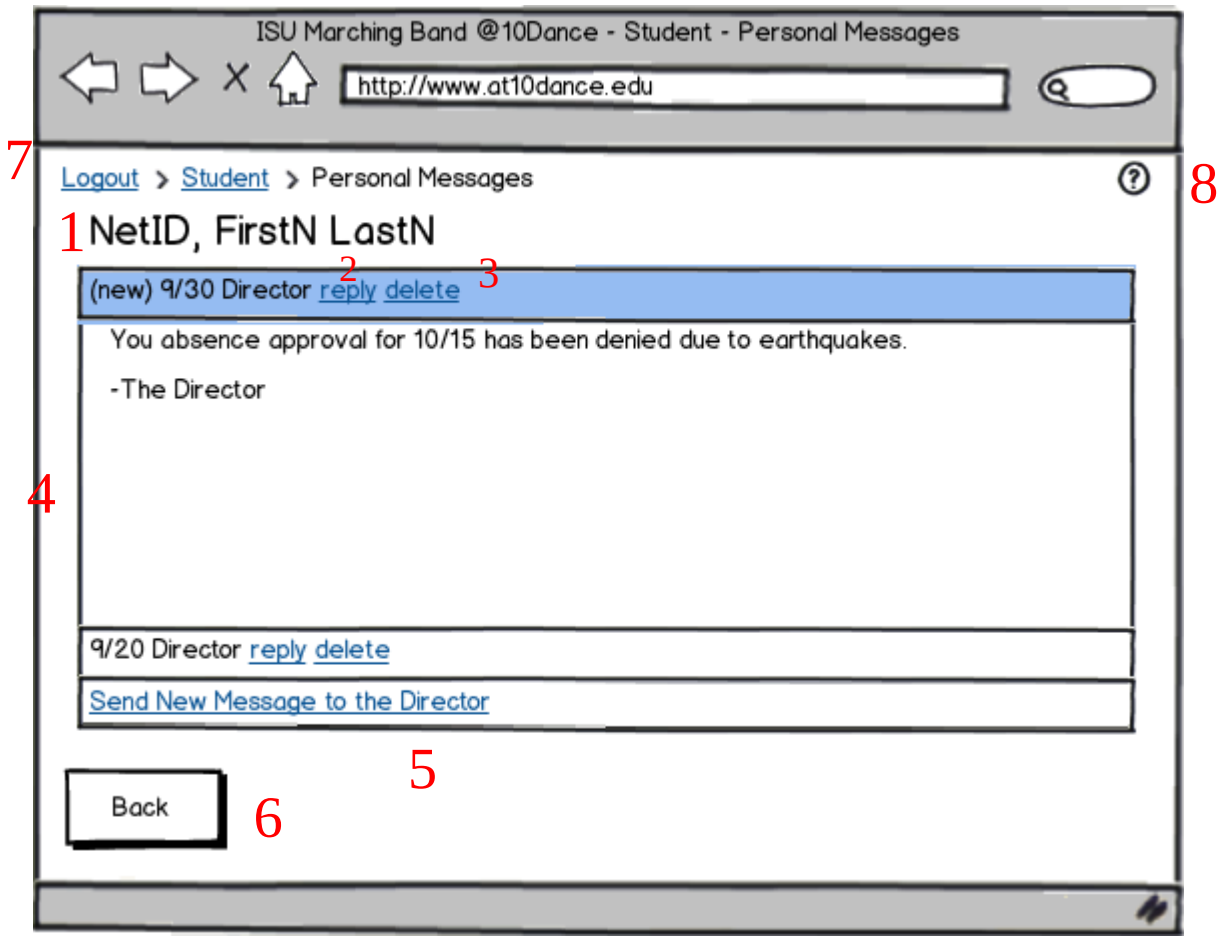
Purpose: This page allows the director to view individual submitted absence forms and approve or deny them.

Loading Events: This page is loaded by clicking the "View" button for an individual absence form in (22_Director_Student_View_Absence) or (23_Director_View_Class_Absence) or by clicking another absence or class conflict form's Next or Prev Button (7)

Interface Elements:
1. Effective Day. Shows the day of the week that this form applies to.

2. Effective Date. Shows the date that this form is valid for.
3. Recurrence Combo Boxes. Shows if this absence will be recurring.
4. Reason Box. Shows the given reason for the absence.
5. Additional Info Box. Shows any additional info given for the absence.
6. View Uploaded Document Button. Downloads the document uploaded with the form, if one was uploaded.
7. Prev/Next Buttons. Allows the user to view the next or previous submitted absence form.
8. Approve Button. Approves the absence form (marks these day(s) as excused).
9. Deny Button. Denies the absence and pulls up the Message Center (33_Director_Message_Center) so the director may notify the student.
10. Back Button. Takes the user back to the List of Student Absence Forms page (29_Student_Absence_Form_List).
11. Page Trail. This allows the user to go back to the Director Main Page (7_Student_Page) or the Student List (15_Director_Page) or the individual Student page (16_Director_Student_List). "Logout" logs the user out and returns them to the main screen (4_Main_Page).
12. Help Button. Pulls up a help page (27_Help) detailing the items on this page and how to use them.
13. Absence Form Label. Shows the student's name, NetID, and the number and type of the absence form the director is viewing.

32_Student_Message_Center
Sketch: Todd Wegter
Annotations: Todd Wegter



Purpose: This is the Student Messaging Center. It allows student the send a personal message to the director and reply to any messages from the director.

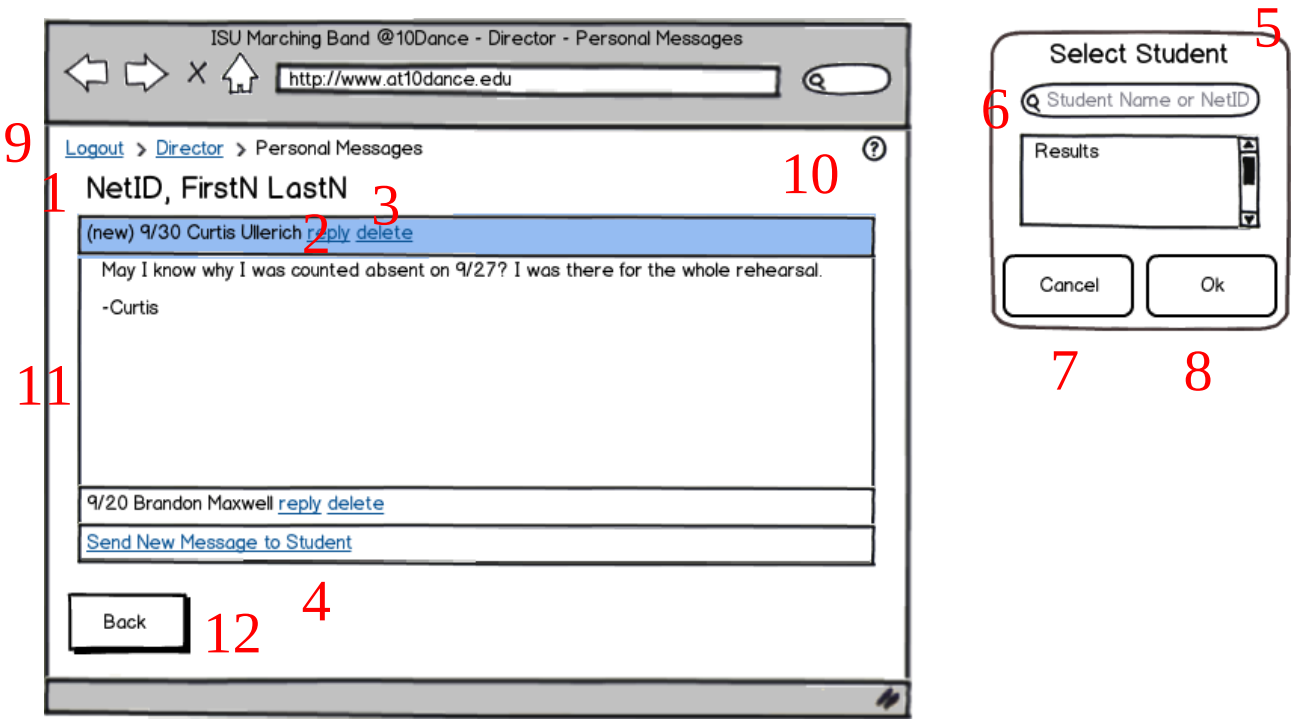
Loading Events: The page is loaded by clicking the "Personal Messages" button in the Student Main Page (7_Student_Page).

Interface Elements:

1. User ID. Shows the User's NetID and First and Last Name.
2. Reply Button. Allows the user to reply to the specified message.
3. Delete Button. Allows the user to delete the specified message.
4. Message Window. Shows headers for each message, allowing each of them to be viewed inside the window.
5. Message Director Button. Allows the student to compose a new message to send to the director.

6. Back Button. Takes the user back to the Student Main Page (7_Student_Page).
7. Page Trail. This allows the user to go back to the Student Main Page (7_Student_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).
8. Help Button. Pulls up a help page (27_Help) detailing the items on this page and how to use them.

33_Director_Message_Center
Sketch: Todd Wegter
Annotations: Todd Wegter



Purpose: This is the Director's Message Center. It allows the director to send a personal message to any student and reply to any message in the inbox.

Loading Events: This page is loaded by clicking the "Personal Messages" Button on the Director Main Page (15_Director_Page) or by denying an Absence Approval Form (31_Director_Student_View_Absence).

Interface Elements:
1. User ID. Shows the User's NetID and First and Last Name.
2. Reply Button. Allows the user to reply to the specified message.
3. Delete Button. Allows the user to delete the specified message.
4. Message Student Button. Allows the director to compose a message to send to any student.

5. Select Student Window. Allows the director to select which student to send a message to.
6. Student Search Bar. Searches the student list for the specified student.
7. Cancel Button. Allows the user to cancel composing a message.
8. OK Button. Allows the user to compose a message to the selected student.
9. Page Trail. This allows the user to go back to the Student Main Page (7_Student_Page). "Logout" logs the user out and returns them to the main screen (4_Main_Page).
10. Help Button. Pulls up a help page (27_Help) detailing the items on this page and how to use them.
11. Message Window. Shows headers for each message, allowing each of them to be viewed inside the window.
12. Back Button. Takes the user back to the previous page, either the Director Main Page(15_Director_Page) or the Absence Form Approval Page (31_Director_Student_View_Absence) or (24_Director_Student_View_Conflict).