LADY LIGHTNING MONKEYS
CHRONOTIMER

Spring 2017
CS 361 Software Engineering

TEAM

Jack Klika

Computer Science & Chinese

David Marx

Computer Engineering

Kaitlyn Montour

Business & Computer Science (minor)

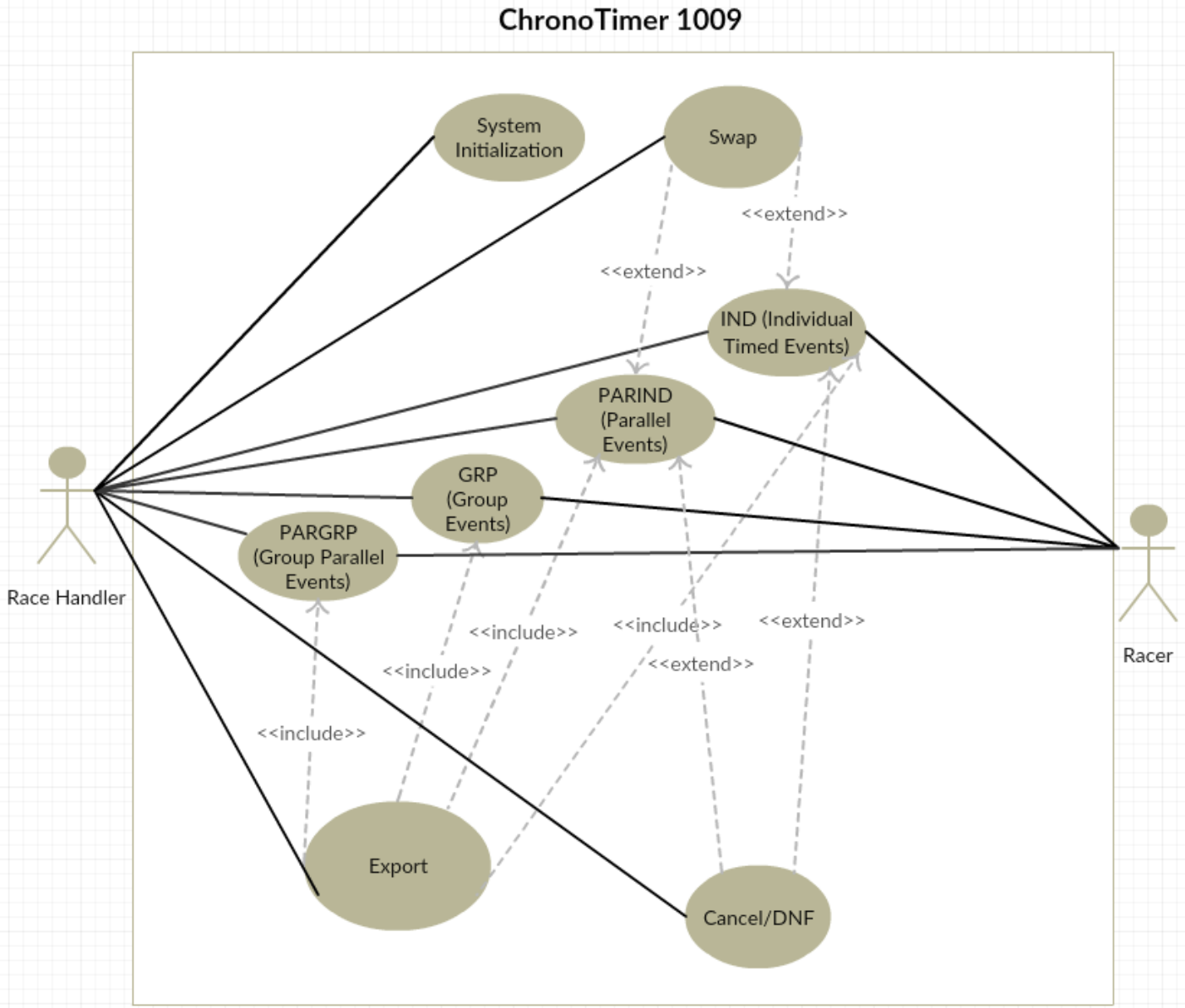
John Aldalali

Computer Science

USE CASES

Race Handler
- Initializes System
- Selects Race Type
- Assigns Bibs
- Connects Triggers to Channels
- Manually scores if needed

Racer
- Gets ready for race
- Crosses start line
- Compete
- Finish first!



FUNCTIONAL REQUIREMENTS

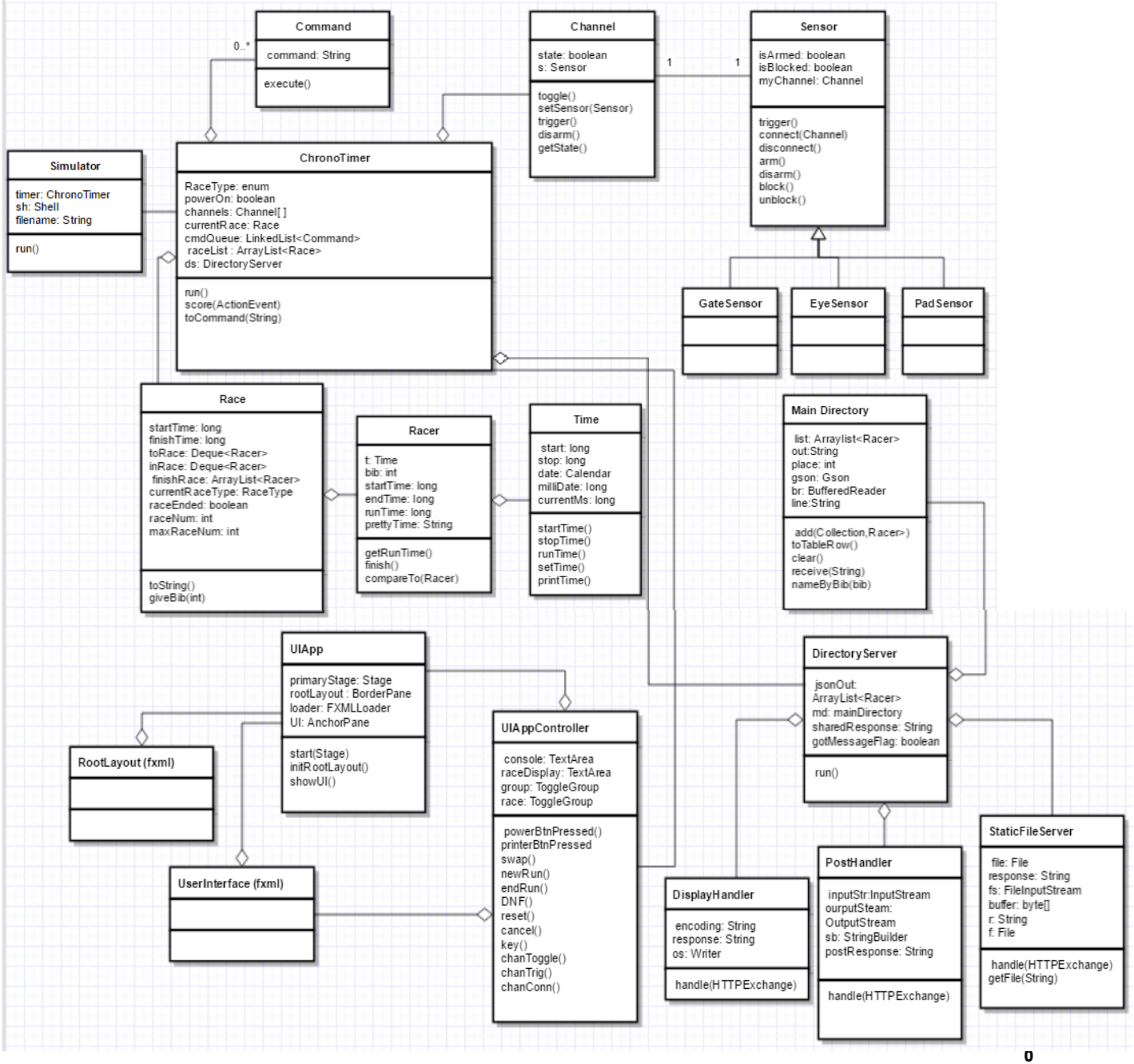
- Timing of sports events by recording start time and end time
  - Live output of race times per racer
- Trigger sensors by variety of methods
  - Pad, gate, or manual trigger
- Handle the following Race Types
  - Individual, Parallel Individual, Group, Parallel Group
- Assign bib numbers to each racer
- Operate system with standard GUI
- Output race results to HTML and USB

NON-FUNCTIONAL REQUIREMENTS

- Sort racers by finish time (1st, 2nd, 3rd, …)
  - Associate bib numbers with racer names via input file
- Usable by multiple race events
  - Swimming, Skiing, Running
- Allow text file input for running commands in sequence
- Printing output gives detailed information of internal system operation
- Records time down to hundredths of a second
- Outputs data in friendly view to user (web server)
- Improper button use does not interfere with race data

DOMAIN MODEL

- ChronoTimer at the center of everything
- Commands used to execute functions in the ChronoTimer
- MVC Pattern used to implement the GUI
- Server directly managed by the ChronoTimer class



TEST PLAN

Chronotimer 1009 Test Plan								
Test Plan ID	**1**	**Test Plan ID**	**2**	**Test Plan ID**	**3**	**Test Plan ID**	**4**
Author	Jack Klika	Author	Kaitlyn Montour	Author	David Marx	Author	John Aldalali
Features being tested		Features being tested		Features being tested		Features being tested	
1	Power on	1	IND Race Start / End	1	PARGRP Race Start / End	1	PARIND Race Start / End
2	GRP Race Start / End	2	NUM	2	DNF	2	Web Server Interface and Functionality
3	Toggle	3	Web Server CSS	3	Web Server Output		
4	CLR	Resources requried:	Chronotimer project files	4	RESET	Resources requried:	Time
5	CANCEL	Testers:	Kaitlyn Montour	Resources requried:	Available time and internet	Testers:	John Aldalali
6	CONN/DISC	Scheduling:	4/25-5/9	Testers:	David Marx	Scheduling:	before 5/9
Resources requried:	Sweat, coffee, and tears	Data/Test files:	CT Project	Scheduling:	4/25-5/9	Data/Test files:	CT Project
Testers:	Jack; maybe his roomates	Initial state:	Program unlaunched	Data/Test files:	CT Project	Initial state:	Program unlaunched
Scheduling:	before 5/9	Components:	Feature sub-features	Initial state:	Program unlaunched	Components:	Feature sub-features
Data/Test files:	CT Project			Components:	Feature sub-features		
Initial state:	Program unlaunched						
Components:	Feature sub-features						

TEST CASES - IND

- Preconditions
  - Power needs to be on
  - Printer is supposed to be on
  - Server should be linked
- IND Specific
  - Click on radio that says IND
  - State the racers
  - Enable and trigger
  - Swap if it applicable
  - State the racers that finish the race
  - DNF the rest that has not finished
  - End race

TEST CASES - PARIND

- Preconditions
  - Power needs to be on
  - Printer is supposed to be on
  - Server should be linked
- PARIND Specific
  - Click on radio that says PARIND
  - State the racers
  - Enable and trigger
  - State the racers that finish the race
  - DNF the rest that has not finished
  - End race

TEST CASES - GRP

- Preconditions
  - Power needs to be on
  - Printer is supposed to be on
  - Server should be linked
- GRP Specific
  - Click on radio that says GRP
  - State the racers
  - Enable and trigger
  - State the racers that finish the race
  - DNF the rest that has not finished
  - End race

TEST CASES - PARGRP

- Preconditions
  - Power needs to be on
  - Printer is supposed to be on
  - Server should be linked
- PARGRP Specific
  - Click on radio that says PARGRP
  - State the racers
  - Enable and trigger
  - State the racers that finish the race
  - DNF the rest that has not finished
  - End race

ITERATION PLANS

Sprint 1	Sprint 2	Sprint 3	Sprint 4	Release Date
Brainstorming	Added PARIND	Added GRP	Added PARGRP	5/9/17
Initializing the structure of the project	Made the system be able to save data	Made an interface to be accessible	Display on the server	
Made some debugs	Handle multiple race types	Print live race output	Corrections and debugs	
Added IND to the system and made a simple start and stop program				

PARTICIPATION

Kaitlyn

- Majority of GUI design and implementation
- Some initial class setup
- Testing
- Diagramming

David

- GUI <--> Simulator interactions
- Primary logic for each race type
  - Scoring, moving racers between queues
- Live race output and UI printing
- DNF, CANCEL, GUI responsiveness
- Many bugfixes

Jack

- Created initial class structure and command execution mechanism
- HTTP Server integration
- Race types and command functionality
- Bugfixes, formatting, and documentation
- Internal debug messages

John

- Initial IND logic
- Channel connection
- RESET

WORKING CODE

Demonstration!