This document describes the current state of Puddles, the Hoofers Online system.

1.0 Completed Uses and Features:

• Create Account
• View Account Info / Membership Info / Ratings
• Modify Account Info
• Change password
• View classes/trips by club and date
• View class/trip detail
• Create class/trip (instructor/CLSU)
• Modify class/trip detail (instructor/CLSU)
• Associate rating requirements with classes/trips (instructor/CLSU)
• Buy a club membership
• Renew a club membership
• Buy/renew membership for another user (clerk)
• Rate club members (instructor/CLSU)
• Search for members of club (CLSU)
• Create membership option
• Edit membership options
• Create rating option
• Edit rating options
• Verify user eligibility for a class/trip
• Register a user for a class/trip (clerk)
• Put a user on a class/trip’s waiting list (clerk)
• Edit member’s volunteer hours (CLSU)
• Promote member to instructor (CLSU)
• Demote instructor (CLSU)
• Promote user to clerk (admin)
• Promote member to CLSU (admin)

2.0 Unimplemented Uses / Features:
• (Un)Suspend user from the system (admin)
• (Un)Suspend member from a club (CLSU)
• Reset another user’s password (clerk/admin?)
• Automatically send users updates about their accounts (password reset, membership expiration, activity cancellation)
• Cancel a trip/class
• Remove user’s registration for a class/trip (CLSU)
• Move a user from an activity’s waiting list to the enrollment list
• Bulk rating changes
• Delete membership option / rating option
• Modify other user’s account info (clerk)
• Buy/renew a membership for another user (clerk)
• Generate membership reports
• Provide anonymous feedback to Hoofers
• Demote a CLSU or clerk
• Promote/Demote an Admin

3.0 Unimplemented Uses / Features Discussed Late in the Project:
• Set up a kiosk like system for the boat house / stable for signing out equipment. We discussed restricting the ability to enroll for an activity by IP address. That field is in the database but is currently ignored. We recommend using Puddles’s DB, permissions scheme, and handlers for checking user eligibility for the equipment ‘activity’. More pages, handlers, and DB tables could be added to Puddles or developed as a separate system for tracking who has a piece of equipment checked out and for maintaining inventory.
• We discussed with Prem the need to change the activity field ‘creator’. It currently is set as the instructor that creates the activity and is used to indicate the leader/contact for that activity. This should be changed to a more general ‘leader’ field and CLSUs should be able to modify the field during/after creation.
• As a feature, we discussed adding two more fields to activity. One is the date to begin displaying the activity (for activities made far in advance). The second is the date to begin allowing enrollment for the activity. The date to end the ability to enroll is already in place and being used.

4.0 Known Bugs / Things to Fix:

• The system has not been checked for compliance with the University’s accessibility requirements
• The disclaimers from the backs of the paper forms are not yet part of the UI.
• The create account page should have a privacy policy for account information. Hoofers should create and display such a policy.
• The create account page does not let the user set the email opt out field or the union membership expiration date.
• Create account does not highlight invalid fields in red.
• The user is currently not allowed to have a null email address.
• Sometimes when displaying prices, the numbers extend out many decimal places even though the DB only stores two decimal places.
• When displaying an activity, the page should also display contact info for the activity’s instructor
• Provide confirmation / summary after successfully creating an activity, creating an account, or promoting a user as an admin.
• There has not been much concurrency testing. As more ‘delete’ type functionality is added to the system, concurrency will become a larger issue.
• Different pages require/display different date formats

5.0 Other Notes:

• You may want to look into finding automated ways to validate that UWIDs (and union ids) entered. DoIT has a system that can do this for non-union ids. Another alternative if you want to enforce UWIDs is to require that members show their ID cards when they have a clerk sign them up for an activity. You could also add a field to the account info that a clerk can set that says that the user has presented the clerk with their ID card. Buy/Renew membership should then check that field and disallow joining if the user has not provided proof of their ID.
• The user manual that we’ve provided could be easily added to the system. That would probably be a good idea.
• If you change the DB schema, you need to regenerate the SQL2Java classes. Some of them (MembershipManager, RatingManager, ActivityRequirementManager) require some bug fixes. Look at what was altered by hand in the save methods and make the same changes.