낚시하시계...
# (Fishing Digital Watch)

Team. 8

201613575 루카이
201712845 류한길
201714168 유호원
201310507 정희찬
201714170 조해성

Contents

001
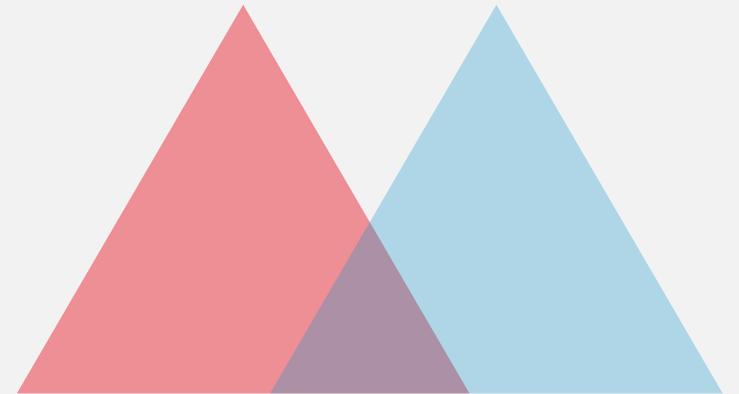
Motivation & Object

Project Introduction
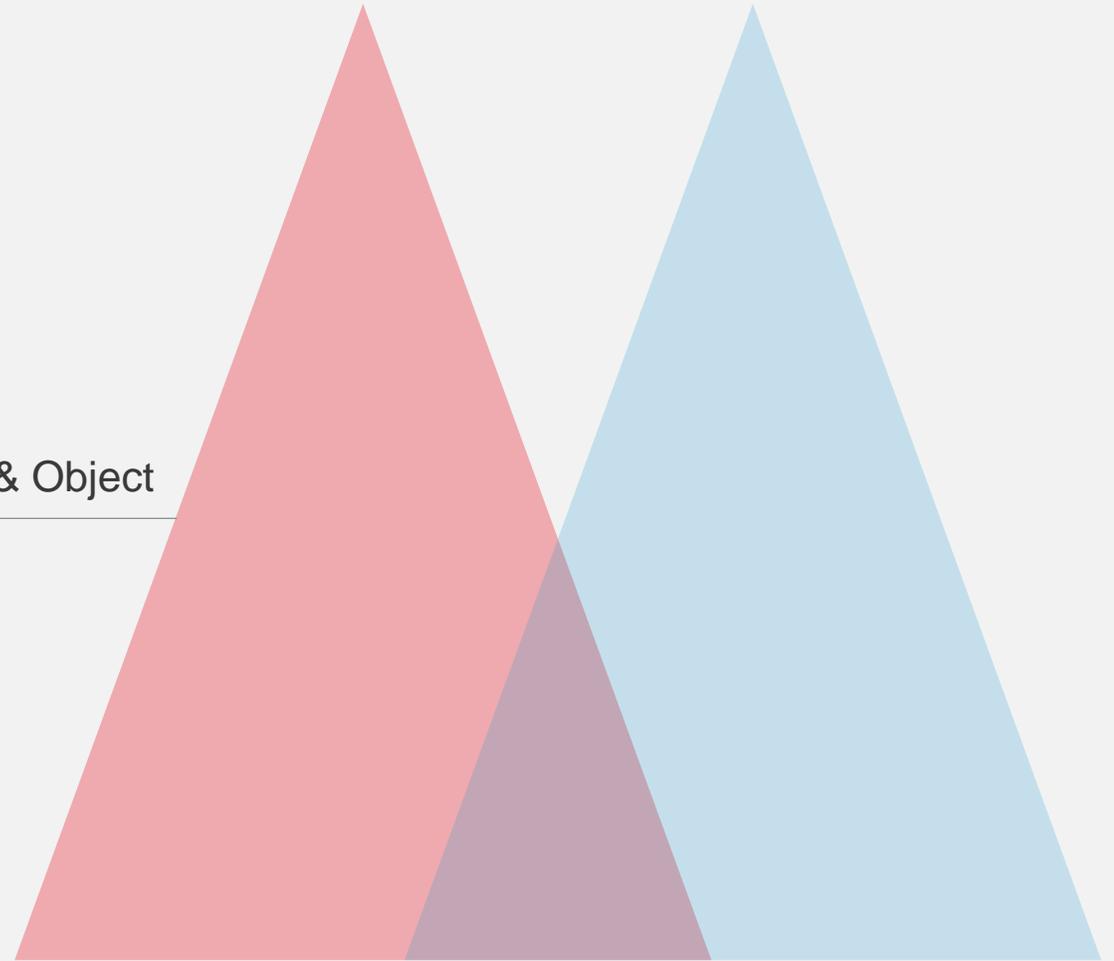
Project Introduction

낚시하시계
(Fishing Digital Watch)

Project Introduction

Alternative solution

외주
맡기기

대학
공모전

낚시
어플
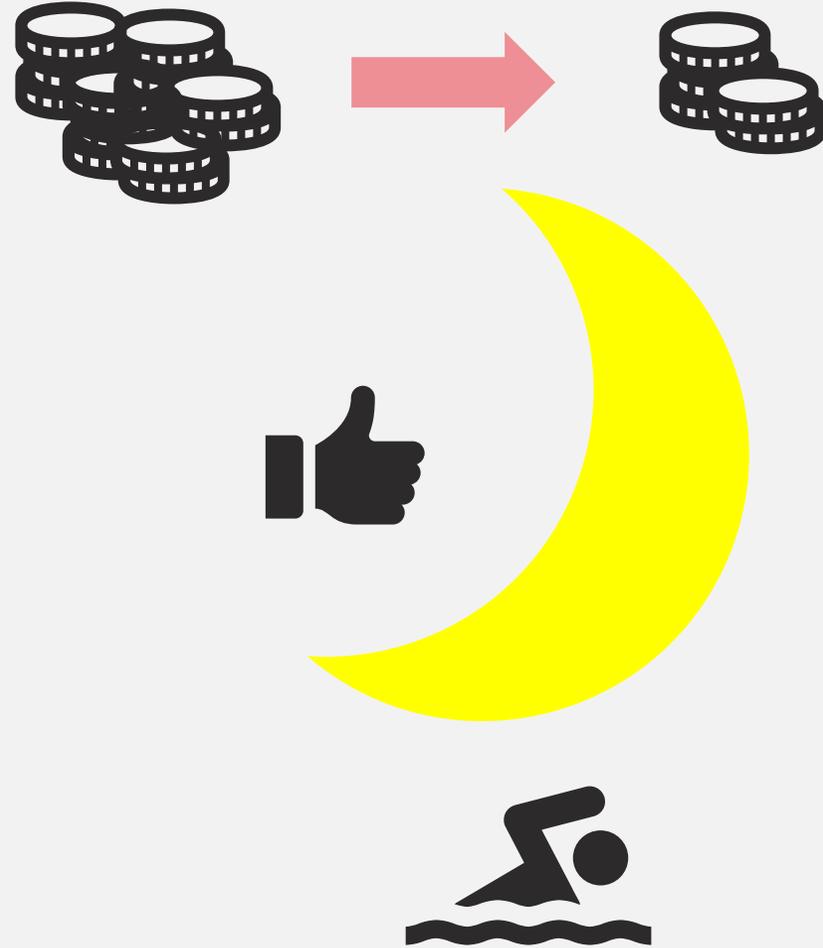
Project Introduction

Human effort : 5M/M
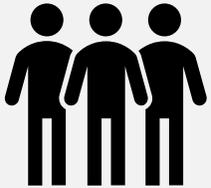
Human resource : 5M

Project duration : 3month
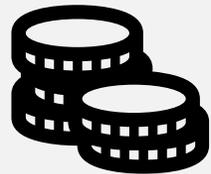
**Cost : 5000만원!**

Project Introduction

Risk	Probability	Significance	Weight	
개발능력부족	4	2	8	공부
의사소통부족	3	1	3	회식
낚시이해도부족	5	1	5	낚시
팀원의 탈주	1	5	5	마음의 편지
팀원 간 시간조율	2	4	8	온라인
맛 없는 학식으로 의지약화	4	5	20	세종대

002

Functional & non-Functional

Requirements

Requirements

time Keeping

set Timer

set Alarm

start Stopwatch

show Tide

show Moonphase

time Adjust

pause Timer

buzz Alarm

record Time

calculate Tide

calculate Moonphase

reset Timer

stop Alarm

pause Stopwatch

next Tide

reset Stopwatch

Requirements

**Simple
is best**

간단하고
직관적인
인터페이스를
제공해야 한다

**Easy
to use**

비 전문가도 사용
하기 쉬워야 한다

**Minimize
Error**

시간계산상 오차를
최소화 해야한다

Requirements

003
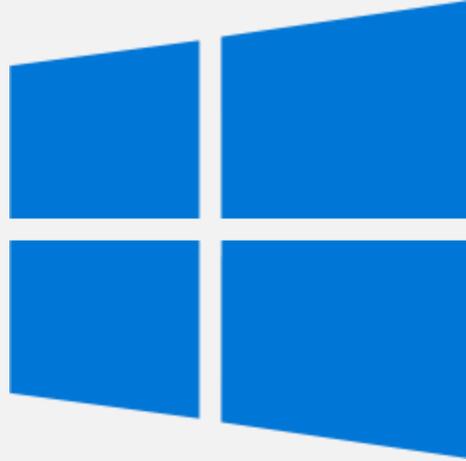
Use-case Diagram

Use-case

Use-case

Watch

User

System
boundary

3

Use-case

Use-case

편리한 UI

저렴한 가격

간편한 휴대성

낚시에 필요한 정보

004
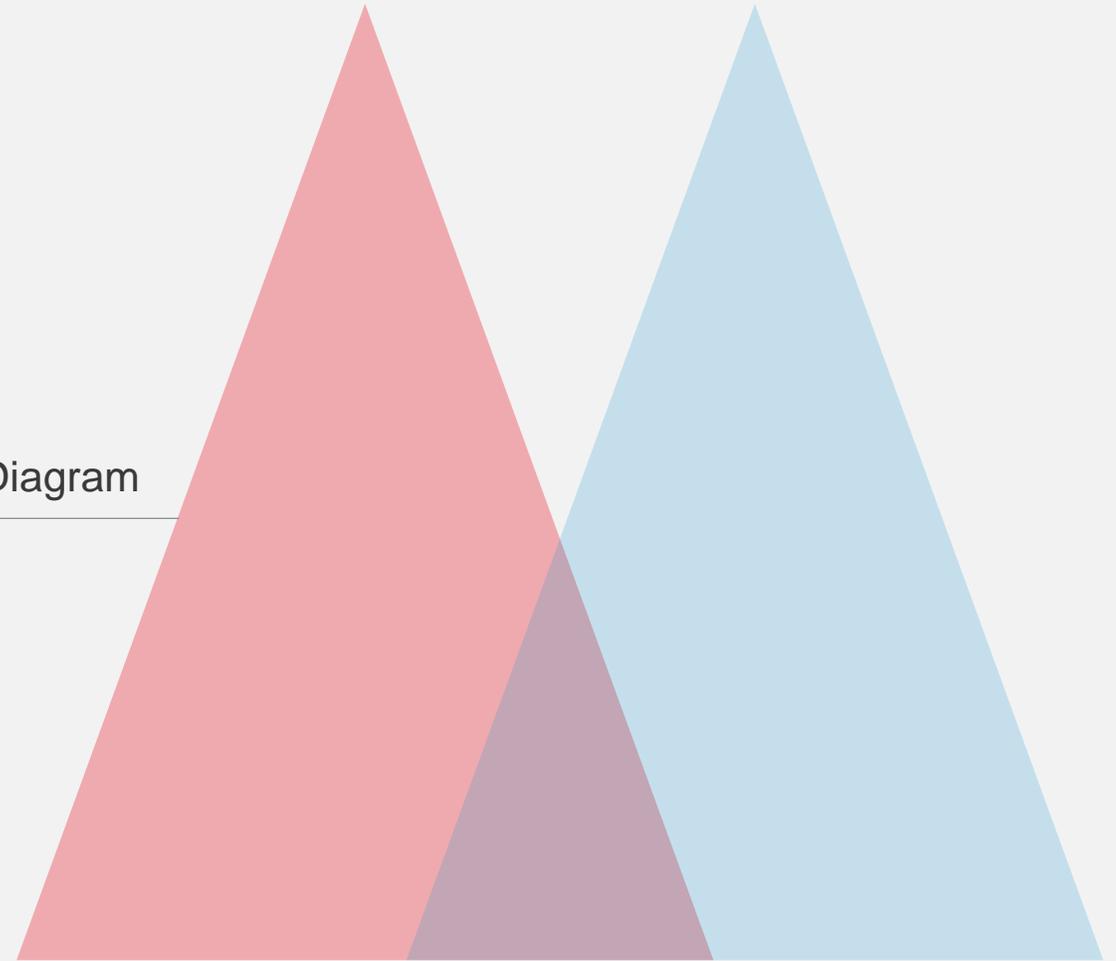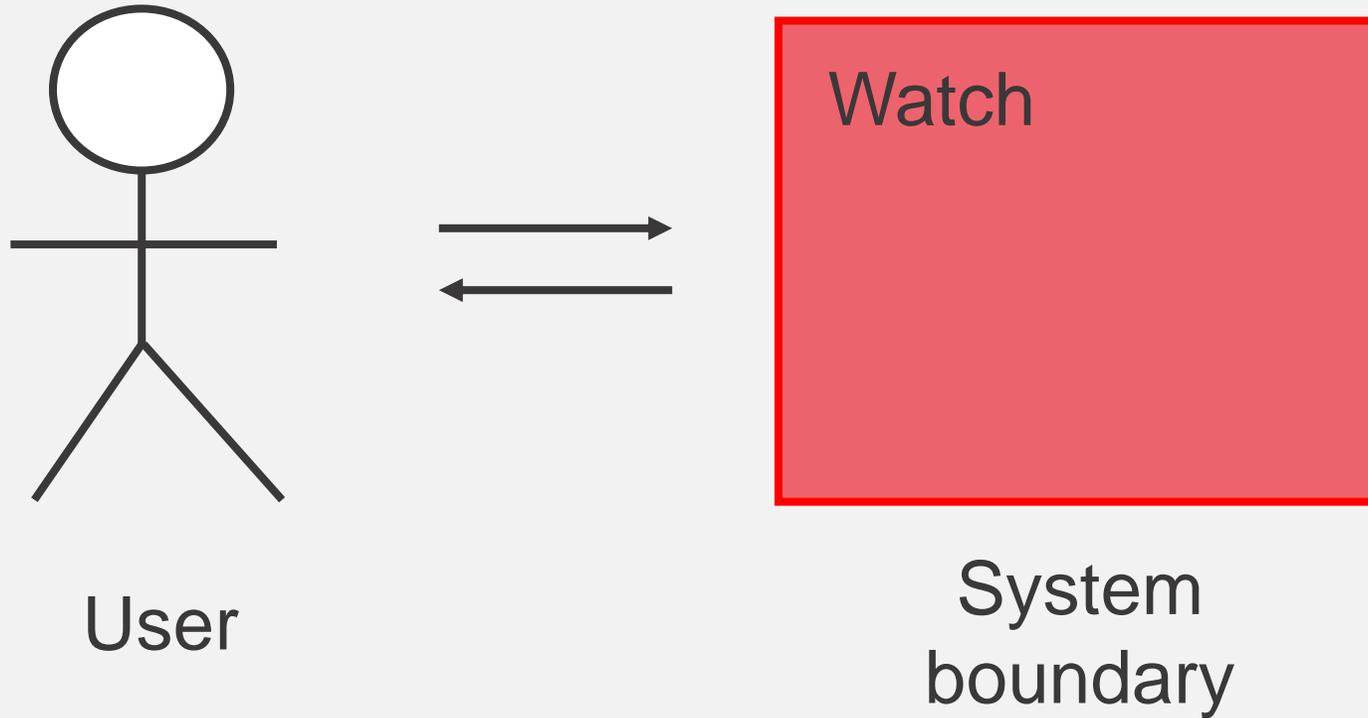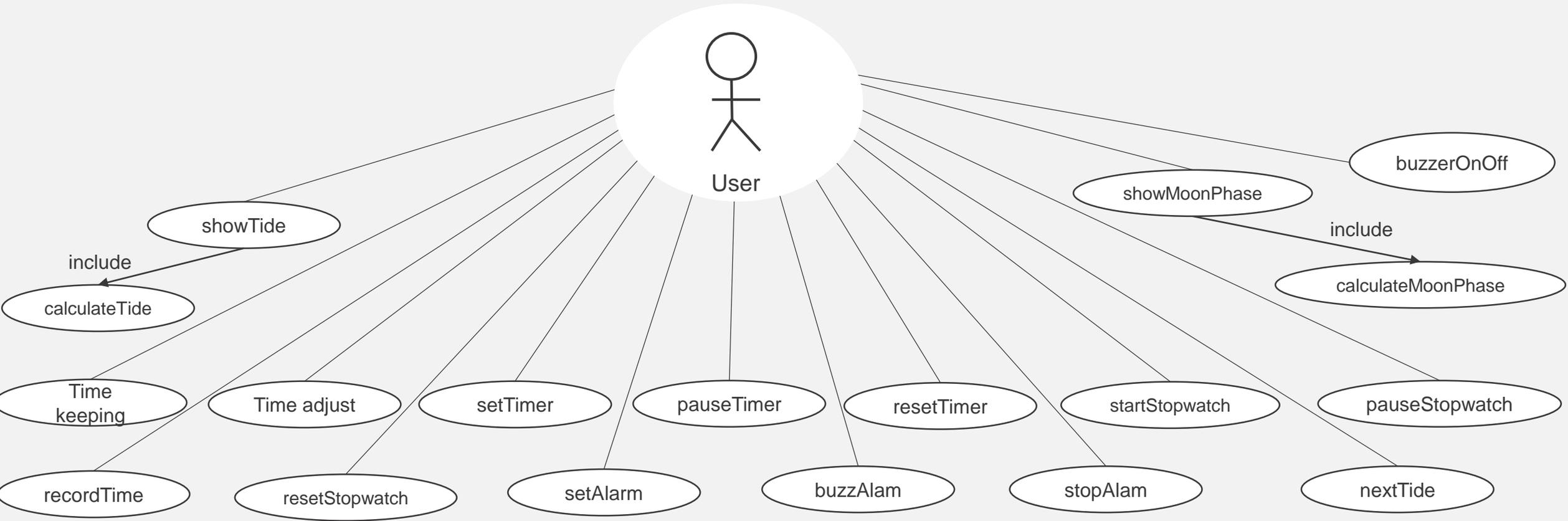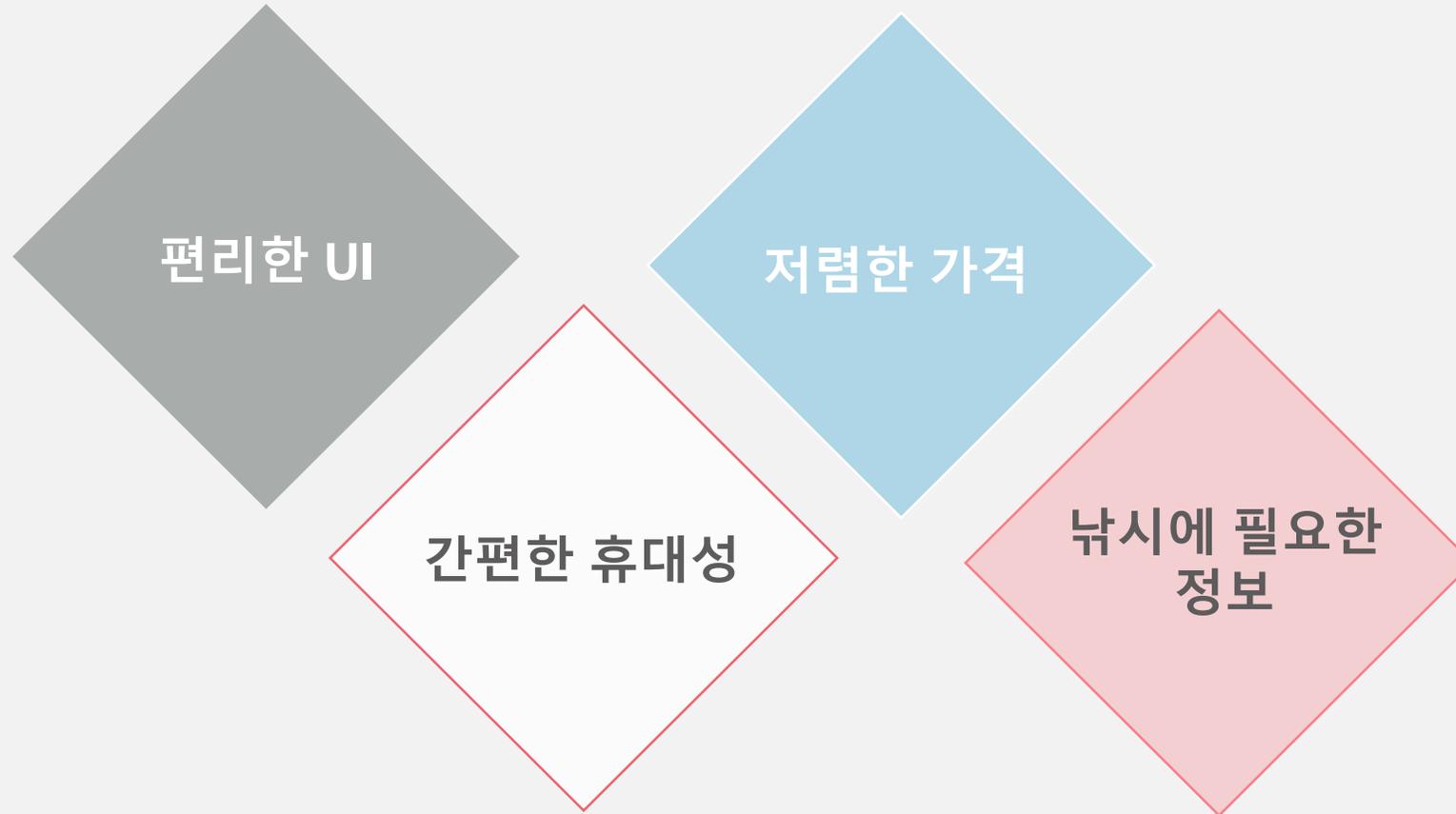
Define system test plan

Test plan

4
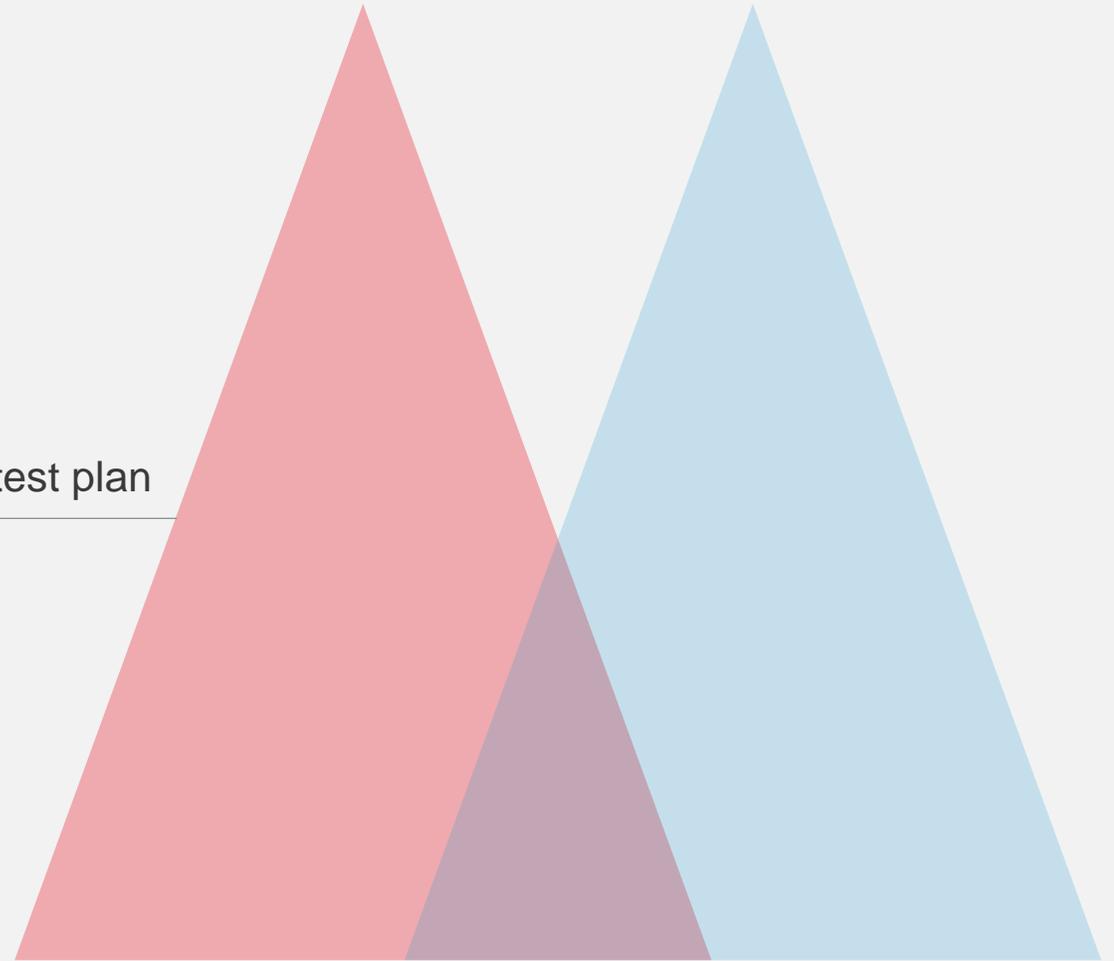
Test plan

time Keeping

set Timer

set Alarm

start Stopwatch

show Tide

show Moonphase

time Adjust

pause Timer

buzz Alarm

record Time

calculate Tide

calculate Moonphase

reset Timer

stop Alarm

pause Stopwatch

next Tide

reset Stopwatch

Scheduling
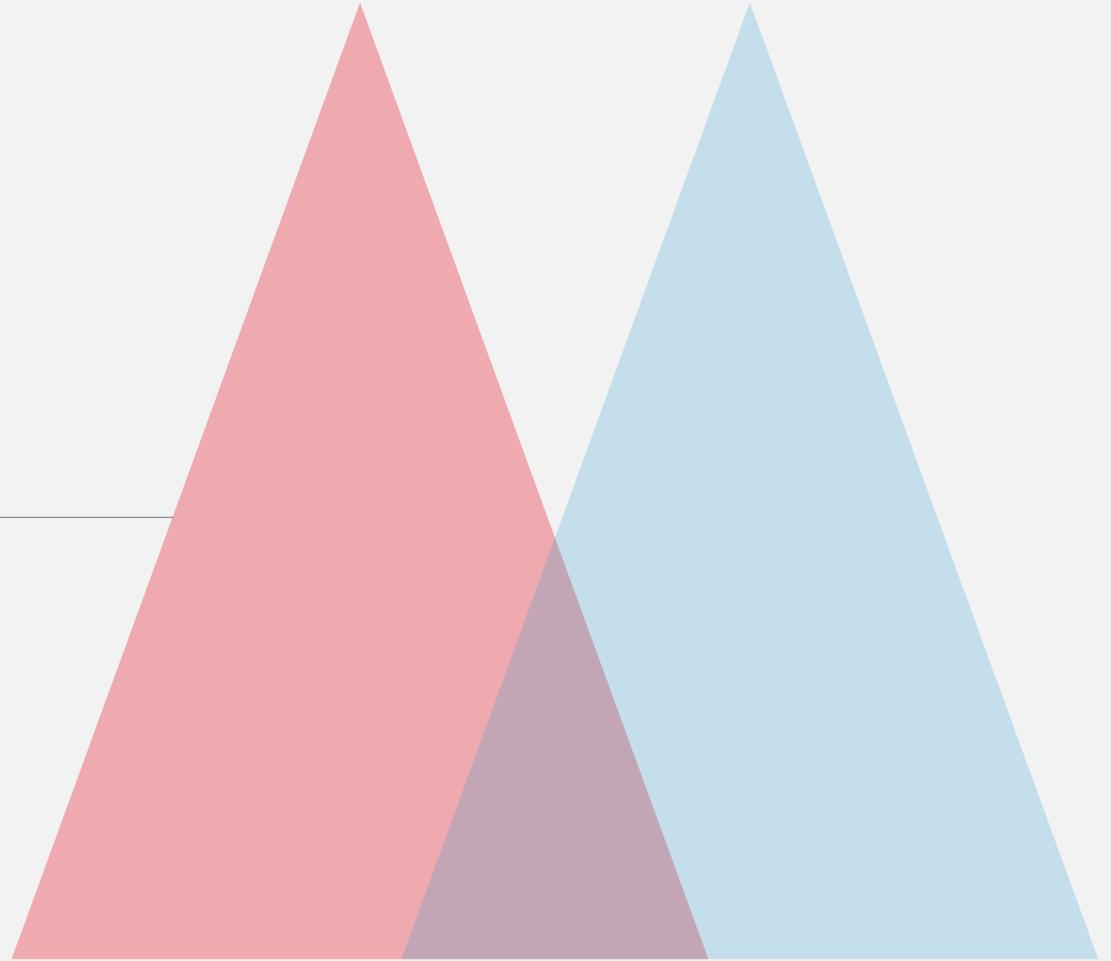
Scheduling

| Activity		schedule (week)										
		1	2	3	4	5	6	7	8	9	10	11
1001. Define Draft Plan		█	█									
1002. Create Preliminary Investigation Report		█	█									
1003. Define Requirements		█	█									
1004. Record Terms in Glossary		█	█									
1005. Implement Prototype		█	█									
1006. Define Draft System Architecture		█	█									
1007. Define Business Use Case		█	█									
1008. Define Business Concept Model		█	█									
1009. Define System Test Case		█	█									
1010. Refine Plan		█	█									
2010. Revise Plan				█	█	█						
2020. Synchronize Artifacts				█	█	█						
2030 Analyze	2031. Define Essential Use Cases			█	█	█						
	2032. Refine Use Case Diagrams			█	█	█						
	2033. Define Domain Model			█	█	█						
	2034. Refine Glossary			█	█	█						
	2035. Define System Sequence Diagrams			█	█	█						
	2036. Define Operation Contracts			█	█	█						
	2037. Define State Diagrams			█	█	█						
	2038. Refine System Test Case			█	█	█						
	2039. Analyze (2030) Traceability Analysis			█	█	█						
2040 Design	2041. Design Real Use Cases						█	█	█			
	2042. Define Reports, UI, and Storyboards						█	█	█			
	2043. Refine System Architecture						█	█	█			
	2044. Define Interaction Diagrams						█	█	█			
	2045. Define Design Class Diagrams						█	█	█			
	2046. Design Traceability Analysis						█	█	█			
	2047. Define Database Schema						█	█	█			
2050 Construct	2051. Implement Class & Methods Definitions									█	█	█
	2052. Implement Windows									█	█	█
	2053. Implement Reports									█	█	█
	2054. Implement DB Schema(SQL, etc.)									█	█	█
	2055. Write Unit Test Code									█	█	█
2060 Test	2061. Unit Testing									█	█	█
	2062. Integration Testing									█	█	█
	2063. System Testing									█	█	█
	2064. Performance Testing									█	█	█
	2065. Acceptance Testing									█	█	█
	2066. Documentation Testing									█	█	█
	2067. Testing Traceability Analysis									█	█	█