StratMaster

A language for algorithmic trading.

The StratMaster Team

Project Manager:	Jintack Lim
Language Guru:	Vincent Mierlak
System Architect:	Xingying Liu
System Integrator:	Moning Zhang
Testing & Validation:	Enrui Liao

The StratMaster Team

Project Manager:	Jintack Lim
Language Guru:	**Vincent Mierlak**
System Architect:	Xingying Liu
System Integrator:	Moning Zhang
Testing & Validation:	Enrui Liao

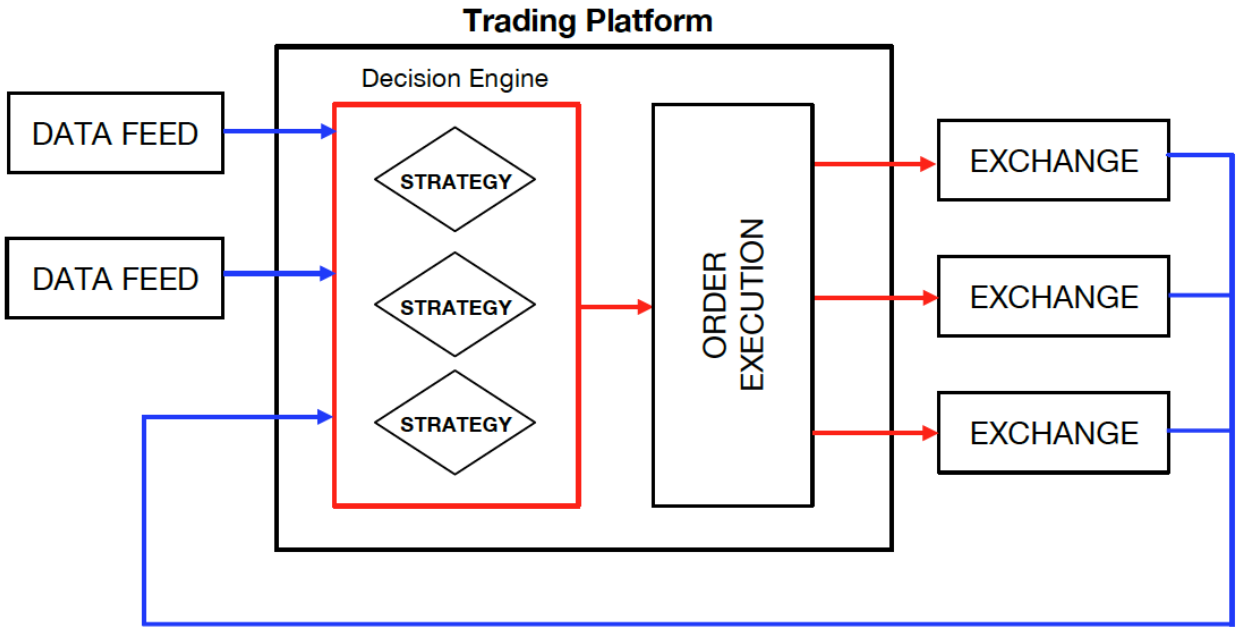
Motivation:

- Computerized trading accounts for over half of all trading volume
- Yet, no domain-specific language for algorithmic trading
- Proprietary software written mostly in C++ (and Java)

Introducing: StratMaster

- Concise and intuitive syntax
- Concurrent
- Modular blocks for algorithms and functions
- Designed for data
- Accounting interface for seamless integration into trading platforms

StratMaster: The Brain Of A Trading Platform



Computational Model

- Process-driven
- Data-driven
- Action-oriented

Demo 1: “Hello World”

```
/* Single Order in a STRATEGY */

USE ACCOUNT ac_master;

STRATEGY hello
{
    BUY { WHAT: SECURITY(EQTY(ZBRA)).AMOUNT(10000).PRICE(USD(85.17)); }
}
```

StratMaster Output: Orders

```
++++++++++++++++++STRATMASTER CONFIRMATION++++++++++++++++++
[2015-05-10 16:16:01] YOU BOUGHT: 10,000 SHARES OF ZBRA AT USD 85.17
 >>>>>> ORDER PLACED BY hello
++++++++++++++++++END CONFIRMATION++++++++++++++++++++++++++

ACCOUNT summary for ac_master:
[  1] ZBRA: Total Shares: 10,000 Current Value:     907,200.00

                Total Value of Securities:          907,200.00
                    Cash Balance Remaining:      99,148,300.00
                                                 ---------------
                       Total Account Value:     100,055,500.00
```

Demo 2: Buy Low, Sell High

- Start with $100 million
- BUY when price < $10
- SELL when price > $15
- Don’t let cash drop below $5 million

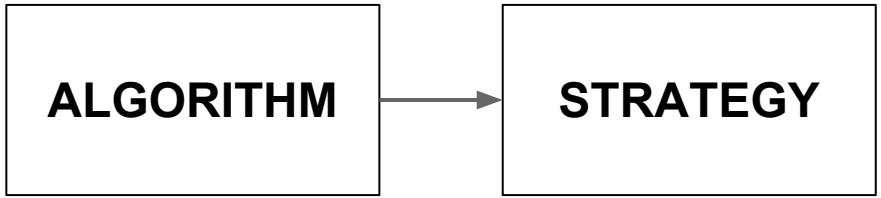
➢ 2 STRATEGYs, 2 ALGORITHMs in parallel

Building Blocks: Types & Objects

Process Blocks:	**Procedures:**
• STRATEGY	• FUNCTION
• ALGORITHM	**Objects:**
USE Elements:	• CURRENCY
• ACCOUNT	• SECURITY
• DATAFEED	• POSITION
• DATABASE	• INT, DOUBLE, PRICE
• EXCHANGE	**Orders: BUY / SELL**

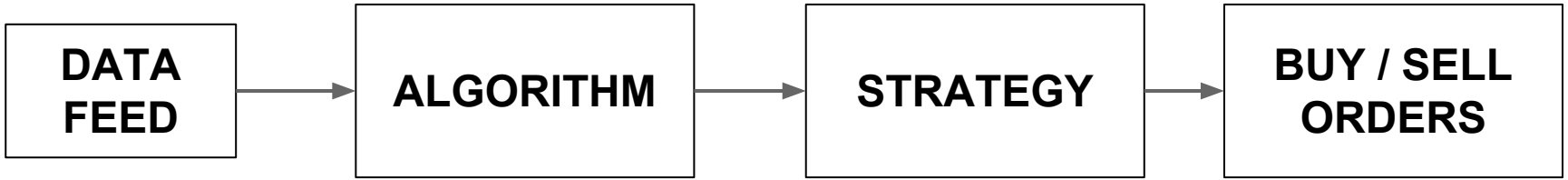
Building Blocks: Types & Objects

Building Blocks: Types & Objects



Building Blocks: Types & Objects

Building Blocks: Types & Objects



Demo 3: Moving Average

- Start with $100 million
- BUY when price > 50-day moving average
- SELL when price < 120-day moving average

➢ 1 STRATEGY with 2 concurrent processes

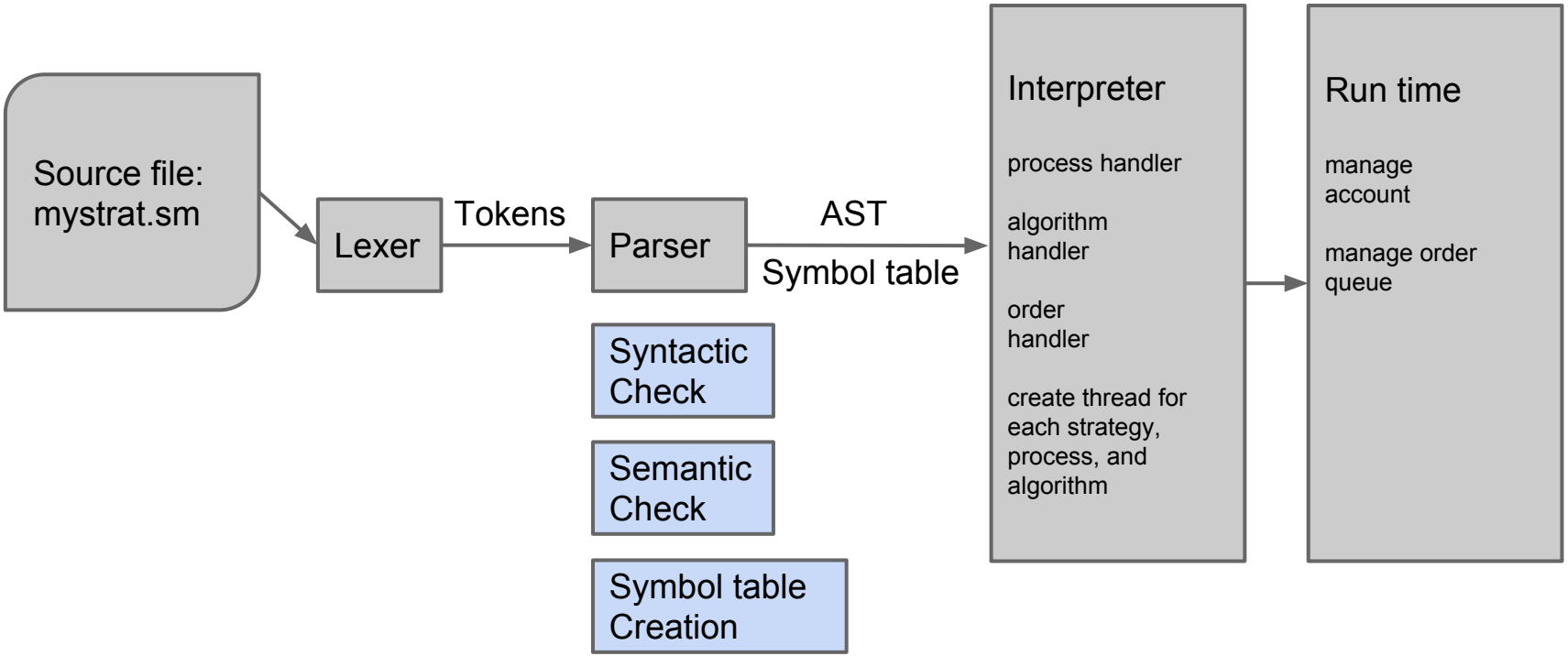
➢ 2 ALGORITHMs

➢ 1 FUNCTION

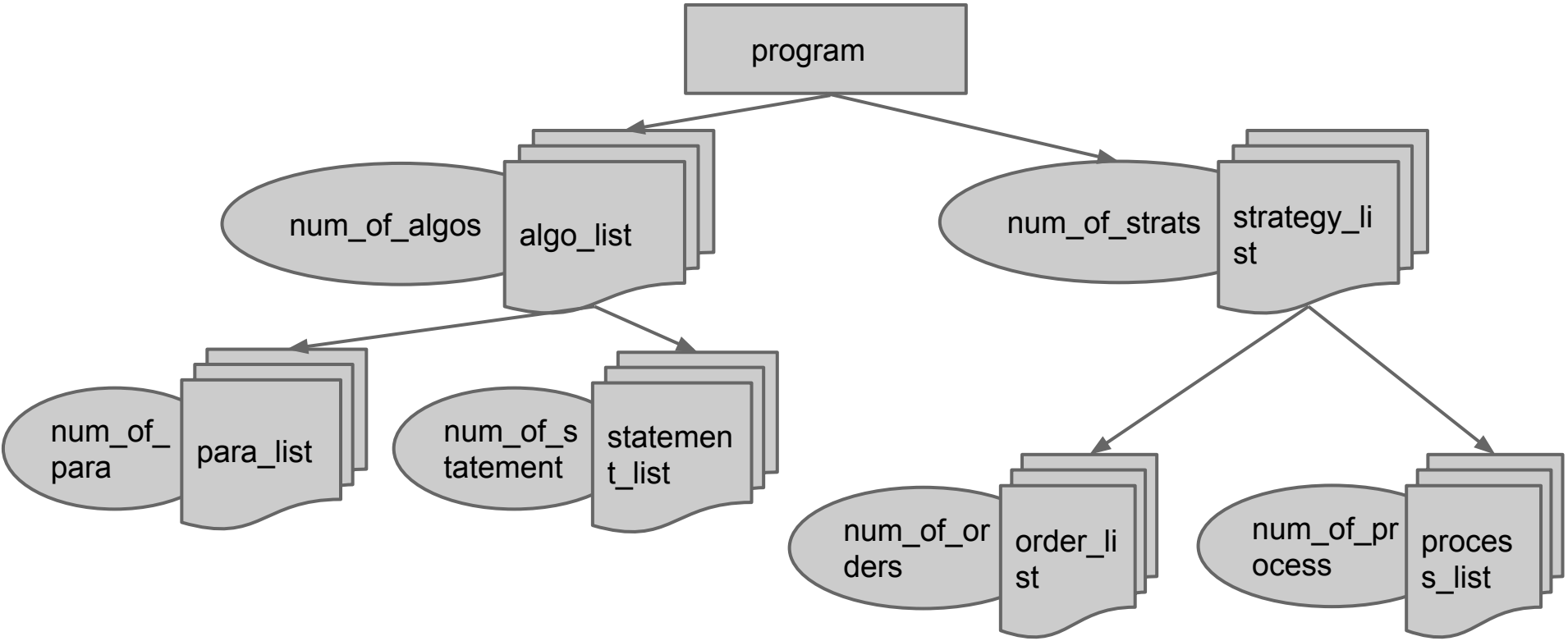
The StratMaster Team

Project Manager:	Jintack Lim
Language Guru:	Vincent Mierlak
System Architect:	**Xingying Liu**
System Integrator:	Moning Zhang
Testing & Validation:	Enrui Liao

Translator architecture



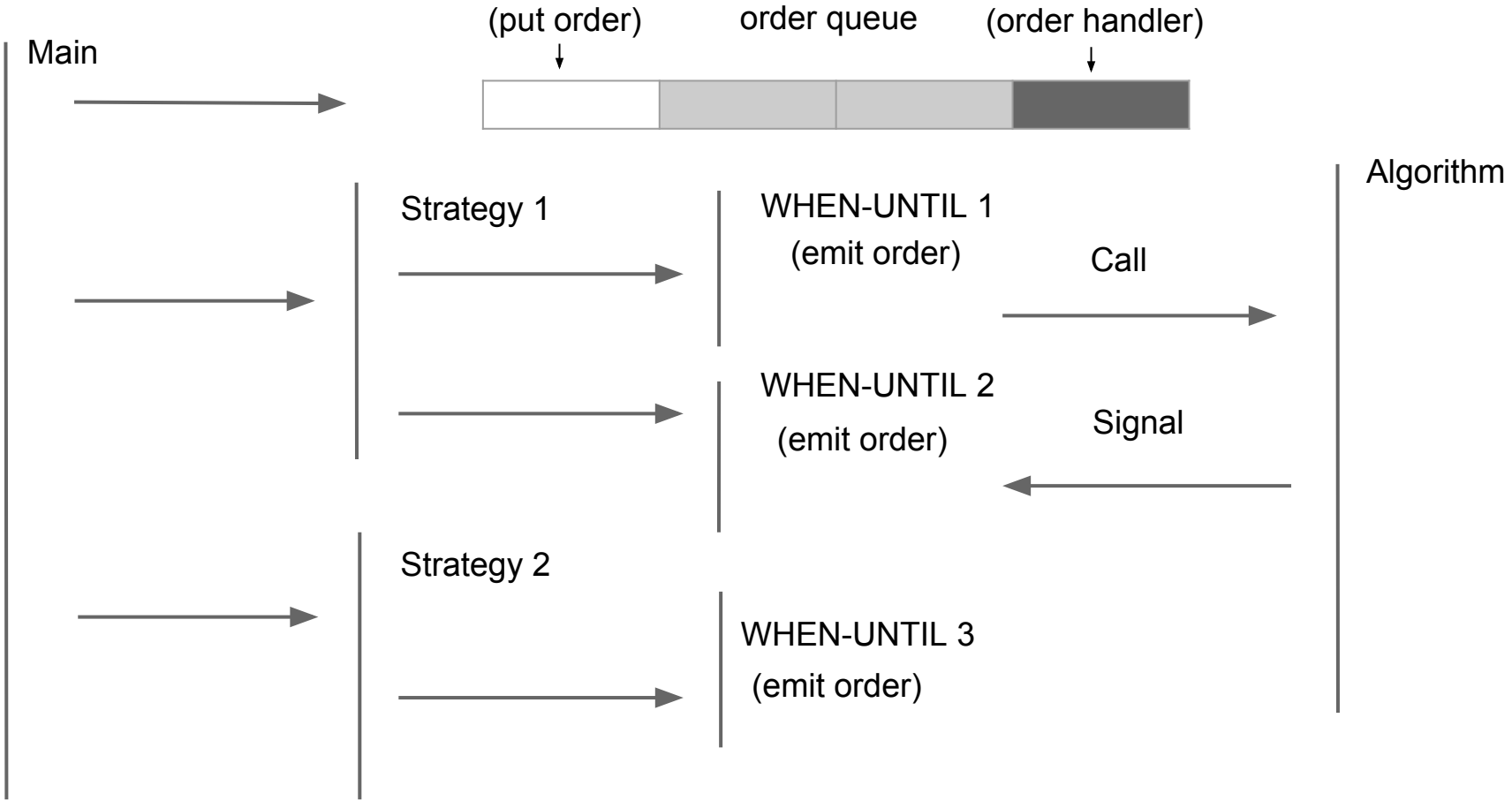
Syntactic Structure



The StratMaster Team

Project Manager:	Jintack Lim
Language Guru:	Vincent Mierlak
System Architect:	Xingying Liu
System Integrator:	**Moning Zhang**
Testing & Validation:	Enrui Liao

Run Time Environment

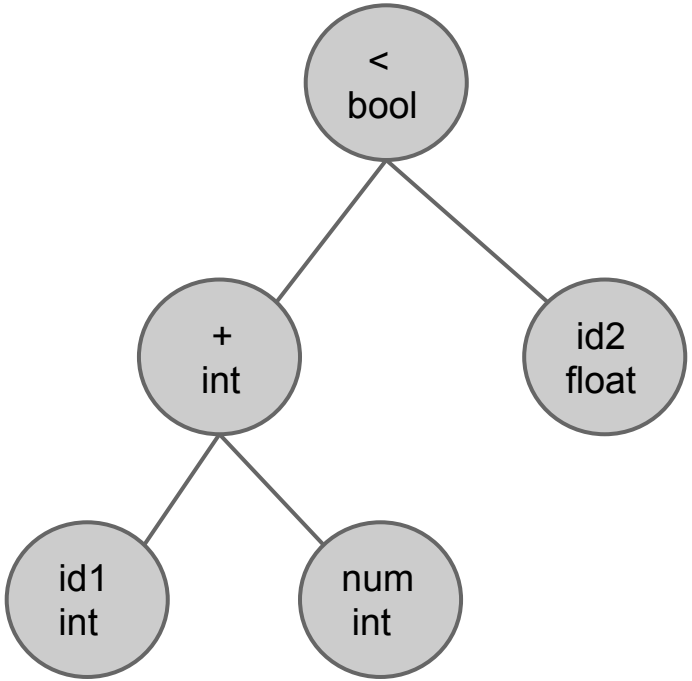


The StratMaster Team

Project Manager:	Jintack Lim
Language Guru:	Vincent Mierlak
System Architect:	Xingying Liu
System Integrator:	Moning Zhang
Testing & Validation:	**Enrui Liao**

Semantic Checking

- What do we do?
  - ID conflict
  - Type error
  - id1 + num < id2
- How we do that?
  - Explicit Stack
  - Type synthesis
  - Postorder traversal



Test Plan:

- What do we check

header: account & datafeed	case 1 - case 15
block: strategy & algorithm	case 16 - case 33
set-if statement	case 34 - case 45
when-until statement	case 46 - case 80
syntax error	case 8-13, case 20, case 52 - case 55
semantic error	case 55 - case 66

Test Plan:

- Interactive test model
  - Partial / Full
  - Log the fail case
  - Easy to debug
  - example

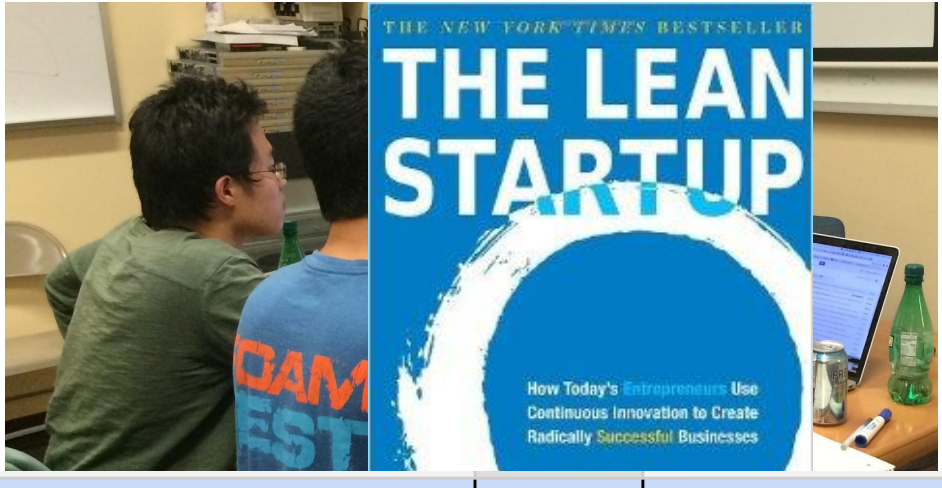
```
case 79 : /*<79>two WHEN statment in one strategy
-------> pass
case 80 : /*<80>two WHEN with until statment in one strategy
-------> pass
/*************summary**************/
pass:  70 / 80
fail case 3 :  /*<3>Missing account
fail case 4 :  /*<4>Account not found
fail case 9 :  /*<9>Wrong Account
```

The StratMaster Team

Project Manager:	**Jintack Lim**
Language Guru:	Vincent Mierlak
System Architect:	Xingying Liu
System Integrator:	Moning Zhang
Testing & Validation:	Enrui Liao

Project Management Strategy

- Team != sum of its parts
- Fast prototyping
- Micro Management



Date	Task	Test file	Status	Person in charge
4/23/2015	multiple orders	misc/progs/orders.sm	Done	Michelle
4/23/2015	algorithm with empty body	devel/emtpy_algo.sm	Done	Jintack, Moning

Conclusions

- what you learned
  - How to work as a team
  - Read and think before you do something

- what you would have done differently
  - To have the more detailed realistic the project plan and keep it strictly

The StratMaster Team

Project Manager:	Jintack Lim
Language Guru:	Vincent Mierlak
System Architect:	Xingying Liu
System Integrator:	Moning Zhang
Testing & Validation:	Enrui Liao

Thank You!